

Hannu Johansson

# TEHTÄVIENHALLINNAN KEHITTÄMINEN TIETOTEKNIKKAPALVELUSSA

Tekniikan ja luonnontieteiden tiedekunta  
Diplomityö  
Maaliskuu 2019

## TIIVISTELMÄ

Hannu Johansson: Tehtävienhallinnan kehittäminen tietotekniikkapalvelussa  
Diplomityö  
Tampereen yliopisto  
Johtamisen ja tietotekniikan DI-tutkinto-ohjelma  
Maaliskuu 2019  
Tarkastajat: professori Marko Seppänen ja Assistant Professori Sami Hyrynsalmi

---

Tietotekniikan palvelualalla yhä useampi yritys pyrkii yksittäisten tuotekehitysprojektien sijaan tuottamaan pitkäjänteisempää palvelua, johon kuuluvat ohjelmistokehityksen lisäksi jatkuvat sovellusten kehitys- ja ylläpitopalvelut. Tällöin yritys pyrkii organisoimaan tuotettavan palvelun ja valitsemaan siihen oikeat tehtävienhallinnan menetelmät varmistukseen tehokkaan palvelutuotannon. Tämän työn tarkoituksena oli tutkia ja kehittää tehtävienhallinnan menetelmiä tietotekniikan palveluyrityksen tietyn asiakkuuden palvelussa. Tutkittavan palvelun tiimeissä tehdään pitkäjänteisen aikataulutetun kehitystyön lisäksi yllätyksellistä ja kiireellistä tuotannon häiriökorjausta. Palvelussa yksittäisen tiimin kehitystyö pitää olla aikataulutettavissa, jonka rinnalla kuitenkin havaitut palvelutuotannon häiriöt tulee korjata nopeasti sopimuksen palvelutason puitteissa.

Tutkittavan palvelun tapaustutkimuksessa toteutettiin teemahaastattelu yrityksen johdolle ja asiantuntijoille. Haastattelun teemat tunnistettiin tukeutumalla kirjallisuuteen asiantuntijatyöstä ja ohjelmistotuotannosta. Haastatteluilla selvitettiin, miten tulisi valita sovel-  
luskehitystä- ja ylläpitoa tekeväille asiantuntijatiimille menetelmät tehtävienhallintaan.

Tutkimuksen tuloksena tunnistettiin hyvin muodostettu tiimi tehtävienhallinnan menetelmien perustaksi ja edellytykseksi tutkittavan palvelun osalta. Tehtävienhallinnan käyttöönotto ja kehitys edellyttävät toimivaa tiimiä, jonka toimintaa tuetaan johtamalla tavoitteiden ja yksilöiden kehittymisen kautta. Tiimin tulee myös tuntea tehtävienhallinnan menetelmät ja jatkuvan parantamisen ajattelu, jotta tiimi kykenee itsenäisesti kehittämään omaa prosessiaan. Lisäksi tehtävienhallinnan menetelmien valintaan vaikuttava tekijä oli rinnakkaiset tehtävävirrat, joissa tehtävävirtojen välisten tehtävien kiireellisyydet vaihtelevat. Tehtävienhallinnan menetelmäksi valittiin tämän johdosta Kanban-menetelmä, jonka keskeisiä ominaisuuksia olivat virtauksen visualisointi, tehtäväkorttien käyttöönotto, keskeneräisen työn rajoitus ja tehtävien palvelutasoluokittelu. Scrum-menetelmän ja muiden ketterän kehityksen menetelmien pohjalta otettiin käyttöön päiväkokoukset, suunnittelukokoukset ja retrospektiivit tiimien prosessien jatkuvaan parantamiseen.

Avainsanat: tietotekniikka, tehtävienhallinta, palvelunhallinta, kanban, agile, lean

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

## ABSTRACT

Hannu Johansson: Development of workflow methods in Information Technology service  
Master of Science Thesis  
Tampere University  
Master's degree Programme in Management and Information Technology  
March 2019  
Examiners: Professor Marko Seppänen and Assistant Professor Sami Hyrynsalmi

---

On Information Technology sector there is the ongoing change that instead of delivering a single software development project companies are trying to provide continuous services which consists of the development and maintenance services. The company will aim to organize the service and choosing the correct methods of workflow management to ensure effective service production.

The motivation of this thesis was research and development of the workflow methods in specific service produced by the case company. In studied service, there were multiple teams that work with scheduled development work but also resolve production incidents that must be fixed quickly according to service level agreement.

The research was conducted as a single case study with themed interviews targeted for the specialist and management roles of the studied service. Themes were found from the literature review of the knowledge work and software production methods. For the use of interviews, the goal was to study how to choose workflow methods for the team that is working with application development and incident management.

As a result of the research, it was confirmed that the foundation for implementing the workflow methods are the well-set teams. Implementing and development of the workflow methods are dependent on performing team. That is reached by leading by setting shared team goals and coaching the team members. The team must have knowledge of the workflow methods and continuous improvement skills to be able to improve independently its own process.

The key for choosing between different workflow methods was identified to be parallel workflows with each different task urgencies. That was a major cause for choosing the Kanban method with features of visualization of the workflow, using the task cards, work-in-progress limit and service levels of the tasks. From the Scrum and other agile methods, there was implementation of the daily stand up, planning meetings and retrospective for the continuous improvement.

Keywords: information technology, workflow methods, service management, kanban, agile, lean

The originality of this thesis has been checked using the Turnitin OriginalityCheck service

## ALKUSANAT

Aloittaessani työuraa keskityin vain ohjelmistokehityksen tekniseen puoleen. Matkan aikana olen ymmärtänyt, että tekniikan lisäksi merkityksellistä työn tekemiselle ja jatkuvuudelle on teknologisen intohimon lisäksi ihmisten välinen kommunikaatio, kannustava johtaminen ja liiketoiminnan arvontuotto. Tuotantotalouden opinnot toivat lisää ymmärrystä ja työkaluja tähän kenttään. Tavoitteeni on jatkossa yhä enemmän yhdistää työssäni näitä näkökulmia ja toivottavasti myös auttaa näiden - toisinaan liiankin - etäisten maailmojen yhdistämisessä.

Kiitokset haluan esittää kaikille prosessissa mukana olleille, erityisesti kohdeyrityksen palvelun henkilökunnalle. Näiden lisäksi erityismaininnan saa Henri Laine mieltä avavasta ketterän toiminnan sparrauksesta ja professori Marko Seppänen erittäin inspiroivasta työn ohjaamisesta. Opiskelun tukijalan mahdollistamisen kiitos kuuluu puolisoleni Marille, joka jaksoi sinnikkäästi pitää kotia yllä, kadotessani toisinaan tehtävienhallinnan kehittämisen alati erittäin mielenkiintoiseen maailmaan. Tämän työn myötä sain toisen ison työn ohella tehdyn opiskeluprojektini päätökseen. Ikuisen oppimisen matka kuitenkin jatkuu – toivottavasti vielä pitkään.

Dubai, 27.3.2019

Hannu Johansson

# SISÄLLYSLUETTELO

1.	JOHDANTO .....	1
1.1	Digitalisaation megatrendistä ohjelmistotuotantoon .....	1
1.2	Tutkimusongelman asettelu ja tutkimuskysymykset .....	2
1.3	Tutkimuksen rajaus.....	4
1.4	Työn rakenne .....	4
2.	KIRJALLISUUSKATSAUS .....	6
2.1	Tietotekniikka-alan yrityksen ja asiakkaan liiketoiminta .....	6
2.2	Asiantuntija- ja tietotyön luonne .....	10
2.3	Asiantuntijoiden yksilö- ja tiimityö .....	11
2.4	Lean toimintastrategiana .....	18
2.5	Perinteisistä suunnittelumalleista kohti ketteryttä .....	23
2.6	Ohjelmistotuotannon viitekehysä ja menetelmiä.....	30
2.7	Yhteenveto tiimityöstä kohti käytännön menetelmiä .....	44
3.	TUTKIMUSSTRATEGIA JA -MENETELMÄT .....	47
3.1	Tutkimusstrategian käytäntöön pano .....	47
3.2	Tiedonkeruu teemahaastattelulla .....	48
4.	TUTKIMUKSEN TULOKSET .....	52
4.1	Tunnistettujen teema-alueiden vastaukset .....	52
4.2	Tehtävienhallinnan käyttöönotto ja kehitys yrityksessä .....	57
4.3	Keskustelua työn tuloksista.....	61
5.	YHTEENVETO.....	63
5.1	Keskeiset tulokset.....	63
5.2	Suosituksat käytäntöön .....	65
5.3	Työn arviointi ja rajoitteet.....	66
5.4	Jatkotutkimuksen aiheita.....	68
	LÄHTEET .....	71
	LIITE 1: MIELLEKARTTA HAASTATTELUN PÄÄTEEMOISTA .....	76

## LYHENTEET JA MERKINNÄT

ICT	Information and Communication Technology, suom. tieto- ja viestintäteknikka
ITIL	Information Technology Infrastructure Library, IT-palvelunhallinnan parhaiden käytäntöjen ohjeistuskirjasto
MVP	Minimum Viable Product, suom. pienin toimiva tuote
PERT	Program Evaluation and Review Technique. Projektinhallinnassa käytetty tilastollinen menetelmä, jossa mallinnetaan monimutkaisen projektin kulua.
PO	Product Owner, suom. tuoteomistaja
WIP	Work In Progress, suom. keskeneräisen työn määrä

# 1. JOHDANTO

## 1.1 Digitalisaation megatrendistä ohjelmistotuotantoon

Digitalisaatio on muuttanut merkittävästi viimeisten vuosikymmenten aikana niin yksityisten kuin yritystenkin arkea. Digitalisaation megatrendinä globaalisti voidaan pitää tietotekniikan laajaa teknologista kehittymistä sekä yleistymistä kuluttajien ja yritysten saataville. Internet on käytössä käytännössä globaalisti, mikä on mahdollistanut nopean viestinnän maapallon laidalta toiselle.

Kehittyneissä maissa kotitalouksilla on jo useita eri tietoteknisiä laitteita, jotka yhdistettynä tietoverkkoon mahdollistavat monia uusia sovellus- ja viestintämahdollisuuksia palveluiden ja kuluttajien välillä. Tietoverkko on laajentunut myös langattomaksi, joten tietoa on saatavilla ja käytettävissä fyysisestä paikasta riippumatta yhä laajemmin. Sosiaalinen media on mullistanut kuluttajan vapaa-ajan käytön, ja täten mahdollistanut myös uusia markkinointimahdollisuuksia. Keskenään yritykset tavoittavat nopeasti toisensa ja talous on muuttunut globaalimmaksi.

Digitalisaation megatrendi ja murros nähdään jo uutena teollisena vallankumouksena aikaisempien höyryn, sähkön ja polttomoottorin tapaan. Digitaalinen vallankumous rinnastetaan kahteen historian suureen talouden vallankumoukseen samoine tuntomerkkeineen. Näiden kummankin seurauksena muuttuivat sekä tuotanto ja jakelutiet: höyry muutti tehdassalit, mutta synnytti rautatieverkon ja höyrylaivaliikenteen. Sähkö hajautti tavaratuotannon ja synnytti liukuhinnat, mutta samaan aikaan kehitetty polttomoottori sekä radio ja lennätin uudistivat jälleen jakelukanavat (Lehti et al. 2012).

Digitalisaatio tulee edelleen muuttamaan yksityisen ja julkisen sektorin organisaatioiden toimintamalleja. Prosesseja on pystytty digitalisaatiolla ja sen tuomalla automaatiolla tehostamaan verrattuna manuaaliseen työhön. Tuotannon toimitusketjusta, tuote- ja palvelutiedonhallinnasta sekä kuluttajakäyttäytymisestä voidaan kerätä yhä laajemmin tietoa. Toisaalta digitalisaation myötä yritys on saattanut luoda kokonaan uusia kumppanuusmalleja, jossa aikaisempi lineaarinen toimitusketju voi muuttua yhä laajemmaksi verkostoksi, jonka hallinta ei olisi ollut aiemmin mahdollista ilman tietoverkon ja -järjestelmien hyväksikäyttöä.

Toisaalta digitalisaatio on myös ravistellut radikaalisti vanhojen toimialojen rakenteita. Tästä yhtenä esimerkkinä finanssi- ja vakuutusala, jossa asiointikanava on siirtynyt yhä enemmän kivijalkaliikkeistä verkkoon. Lisäksi globaalimpi talous ja säätely on avannut

kilpailua kevyemmän rakenteen toimijoille kotimaan rajojen ulkopuolelta, ja lisännyt täten toimialan kilpailupainetta, koska digitalisaation myötä yritykset ovat tavoitettavissa yli rajojen.

Uudet digitalisaation mahdollistamat liiketoimintamallit saattavat olla hyvinkin poikkeavia aiemmista niin rakenteiden kuin voitonjaon suhteen. Tulevaisuuden kasvavina digitalisaation osa-alueina nähdään alustatalous, tekoälyn kehittyminen, virtuaalitodellisuuden sovellusten käyttöönotto ja langattoman tiedonsiirron suorituskyvyn ja ulottuvuuden parantuminen, teollinen internet eli yhä useamman laitteen kytkeytyminen verkkoon tuottamaan tietoa sekä yhä autonomisemmat ratkaisut. Kaikki tämä nähtävä kehitys tarkoittaa sitä, että tietotekniikan ohjelmistoliiketoiminta ja -tuotanto tapojenkin täytyy kehittyä (Metsä-Tokila 2017; Press 2017).

Täten jokaisen yrityksen toimintaprosessit tulevat aina yhä enemmän riippuvaiseksi tietoteknisistä ratkaisuista. Teknologioiden määrän lisääntyessä ja monimuotoistuessa, myös tietotekniikkainfrastruktuuri kerrostuu ja monimutkaistuu, ja tarve tähän liittyvälle tietotekniikan palvelulle tulee korostumaan. Tätä tarvetta ovat tyydyttämässä tietotekniikan palvelualan yritykset.

Palvelutuotannon määrittelyyn Informaatio- ja viestinnän (ICT) toimialalla sisältyy esimerkiksi erilaisten tuotteiden, laitteiden, konsultaatioiden, ratkaisujen tai palvelujen myymistä. Hyvänä esimerkkinä yrityksistä, jonka päätoimiala luokitellaan muuksi kuin ohjelmistopalvelut, voivat olla yritykset, joiden tuotanto on hyvin teknologiaan painottuva. Esimerkkinä suomalainen teleoperaattori, joka kehittää vahvasti ohjelmistoja tuotekehityksenä tuotantonsa tarpeisiin tai laitevalmistaja, esimerkkinä aikanaan Nokian matkapuhelinliiketoiminta. Näissäkin yrityksissä ohjelmistopalvelun tuottaminen sisäisiin tarpeisiin on silti hyvin vahvasti edustettuna, vaikka ne eivät kuulu ohjelmistopalvelujen luokitukseen. Tässä työssä keskitytään ohjelmistopalveluun, jota tietotekniikan palveluyritys voi tuottaa asiakkaidensa tai oman liiketoimintansa ratkaisutarpeisiin.

## **1.2 Tutkimusongelman asettelu ja tutkimuskysymykset**

Tämän työn lähtökohtana oli tietotekniikan palveluyrityksen palvelu- ja ohjelmistoliiketoiminnan kehittäminen asiantuntijatyöhön liittyen. Yritys toimittaa useille asiakkaille pääasiassa yhden tai useamman tiimin laajuisia ohjelmistojen kehitys- ja toimitusprojekteja.

Projektit koostuvat tyypillisesti yrityksen itse kehittämien sovellusratkaisujen tuottamisesta tai kumppaniyritysten valmissovellusten konfiguroinnista asiakkaan tarpeiden mukaan. Ohjelmistotuotannon toimialalla puhutaan tavoitteesta tarjota kokonaisratkaisuja ns. jatkuvana palveluna. Tämä tarkoittaa varsinaisen ohjelmistoprojektin sovellustoitumisen lisäksi palvelusopimusta, joka kattaa myös jatkossa sovellusratkaisun ylläpito- ja muutostyöt määräajaksi eteenpäin. Tämän liiketoimintamallin motiivina on tarkoitus



saada tasaista ja ennustettavaa liikevaihtoa, sen lisäksi, että myytäisiin pelkkiä yksittäisiä ohjelmiston toimitusprojekteja.

Jatkuvan palvelun tuottamisen lähtökohtana on henkilöiden ja työnohjauksen järjestäminen niin, että palvelutuotannon palvelutasot kyetään pitämään sopimusrajoissa ja toisaalta tämän rinnalla tehtävän sovelluskehityksen aikataulut eivät häiriinny. Haasteita toimivan mallin löytämiseen tuo esimerkiksi palvelutuotannon häiriöiden ennustamattomuus, jolloin asiantuntijoiden pitää sopimusaikojen puitteissa analysoida ja korjata tuotannon häiriöt. Tämän rinnalla saman tiimin asiantuntijat tekevät myös sovelluksiin liittyvää kehitystä. Sovelluskehitykseen kohdistuu tyypillisesti kiinteät aikatauluvaatimukset, jotka eivät saa taas liaksi kärsiä palvelutuotannon työstä. Kehitettäviä tai muokattavia sovelluksia saattaa tiimillä olla useita. Liiketoimintamallissa kehitys on tyypillisesti laskutettavaa tuntityötä, mutta palvelutuotanto kiinteähintainen sopimusjakso, joten yrityksen tavoitteena on kohdistaa henkilöresurssit mahdollisimman tehokkaasti laskuttavaan työhön, esimerkiksi ohjelmistokehitykseen tai -konfigurointiin. Toimittaessa saman asiakkuuden palveluprojektissa on myös tärkeää, että osaaminen kehityksen- ja palvelutuotannon välillä kehittyy. Tämän takia on usein mielekästä, että samat henkilöt samassa tiimissä hoitavat kehityksen- ja palvelutuotannon, joten tiimien tehtävienhallinta on tämän kaiken keskiössä.

Tässä työssä vertaillaan tietotyön sekä erityisesti ohjelmistotuotannon vakiintuneita viitekehyksiä ja käytäntöjä, joista pyritään löytämään keskeisimmät ominaisuudet tutkittavan kohdepalvelun käyttöön. Ominaisuuksien valintaperusteita selvitetään tutkimalla menetelmiä kirjallisuudesta ja haastatteleamalla palvelun henkilöstöä. Yritys, palvelu, asiakas ja sen asiantuntijat muodostavat aina yksilöllisen toimintaympäristön, joten haastatteleamalla tämän yksittäisen palvelun henkilöitä, pyritään löytämään keskeisimmät valinnat juuri tälle palvelulle.

Onnistuneilla valinnoilla pyritään parantamaan palvelutuotannon tehokkuutta ja hallittavuutta sekä tarjoamaan asiantuntijatiimeille selkeämmät menetelmät moniulotteisessa työympäristössä. Parhaimmillaan onnistuneella tehtävienhallinnan menetelmien valinnalla tuottavuus nousee, jonka lisäksi ehkäistään henkilöiden ylikuormitusta, ja saavutetaan työn tekemiseen terveys- ja motivaatiohyötyjä.

Tässä tutkimuksessa tutkittavan kohdepalvelun osalta pyritään etsimään tehtävienhallintaan vaikuttavia tekijöitä, joiden avulla palvelua voitaisiin parantaa. Tutkimuksen kanalta ongelman asettelu on muotoiltu seuraavaksi päätutkimuskysymykseksi:

- *Miten valita sovelluskehitystä ja -ylläpitoa tekeville asiantuntijatiimille tehtävienhallinnan menetelmät?*

Jotta varsinaiseen tutkimuskysymykseen voidaan vastata, on kysymys jaettu kahdeksi alikysymykseksi:

- *Mitkä tekijät vaikuttavat tehtävienhallinnan menetelmien valintaan?*
- *Mitkä tehtävienhallinnan menetelmien ominaisuudet valittiin käyttöön?*

Tutkimuksen tavoite on tutkia ja parantaa yrityksen palvelua tehtävienhallinnan osalta. Kohdepalvelun tehtävienhallinnasta yritetään tunnistaa keskeiset asiat, jotka tutkittavassa kontekstissa vaikuttavat asiantuntijatiimin tehtävienhallintaan ja löytää niiden pohjalta tehtävienhallinnan olennaisimmat ominaisuudet. Löydettyjä asioita peilataan alan tunnettuun kirjallisuuteen ja tutkimustietoon, jonka pohjalta tehdään johtopäätöksiä kehitystoimenpiteistä.

### 1.3 Tutkimuksen rajaus

Tämä tutkimus rajattiin tietotekniikan palveluyrityksen yksittäiseen asiakkaalle tuotettavaan palveluun. Tutkimuksen taustavaikuttimena oli halu kehittää yrityksessä juuri tämän palvelun toimintaa ja lisätä kehittämisen myötä myös osaamista tehtävienhallinnan menetelmiin.

Tehtävienhallinnan menetelmien valintaan vaikuttavat tekijät voivat myös olla hyvin yksityiskohtaisia riippuen toimintaympäristöstä, joten yleistävä tutkimus ei olisi välttämättä palvellut riittävästi yrityksen kohdepalvelun kehittämistä.

Tehtävienhallintaan vaikuttavien tekijöiden tutkinnan osalta rajaus pyrittiin tekemään mahdollisimman lähelle tutkittavien tiimien näkökulmaa. Tehtävienhallinnan osalta rajattiin pois projektin- ja palvelunhallintamenetelmien tutkiminen, laadunparannus, yksilötason työhyvinvointi, yritysten organisaatorakenteet, liiketaloudelliset muuttujat ja asiakasyrityksen toiminta.

### 1.4 Työn rakenne

Työ on jaettu viiteen päälukuun. Ensimmäisenä tämä luku johdattelee ongelmaan ja kuvaa tiivistä tämän työn sisällön kokonaisuutena.

Toinen luku on katsaus alan kirjallisuuteen ja tutkimuksiin. Luku kuvaa ensiksi lyhyesti tietotekniikan liiketoiminnan ohjelmisto- ja palvelutuotannon osalta. Tämän jälkeen kuvataan tietotyön asiantuntijoiden yksilö- ja tiimityön piirteitä, jonka jälkeen esitellään ohjelmistotuotannon yleisimmät prosessi- ja työnohjaukseen liittyvät menetelmät niiden taustoituksineen.

Kolmannessa luvussa kuvataan tutkimuksen strategia, jossa perustellaan miksi valittiin tutkimusmenetelmäksi tapaustutkimus teemahaastatteluilla. Luku kuvaa miten haastattelun pääteema-alueet on valittu ja miten haastatteluiden tulokset ovat analysoitu.

Neljäs luku esittelee tutkimuksen haastattelusta tehdyt vastaustiivistelmät, tulkinat ja teoriasta johdetut kehitystoimenpiteet tiimien tehtävienhallinnan tueksi yritykselle. Lopuksi käydään keskustelu työn tuloksista verraten muihin vastaaviin tutkimuksiin.

Viides luku tiivistää yhteenvedona työn sisällön ja tutkimuksen tulokset. Luvussa annetaan suositukset kohdepalvelun kehittämiseen, esitykset jatkotutkimukseen ja arvioidaan työtä kriittisesti.

## 2. KIRJALLISUUSKATSAUS

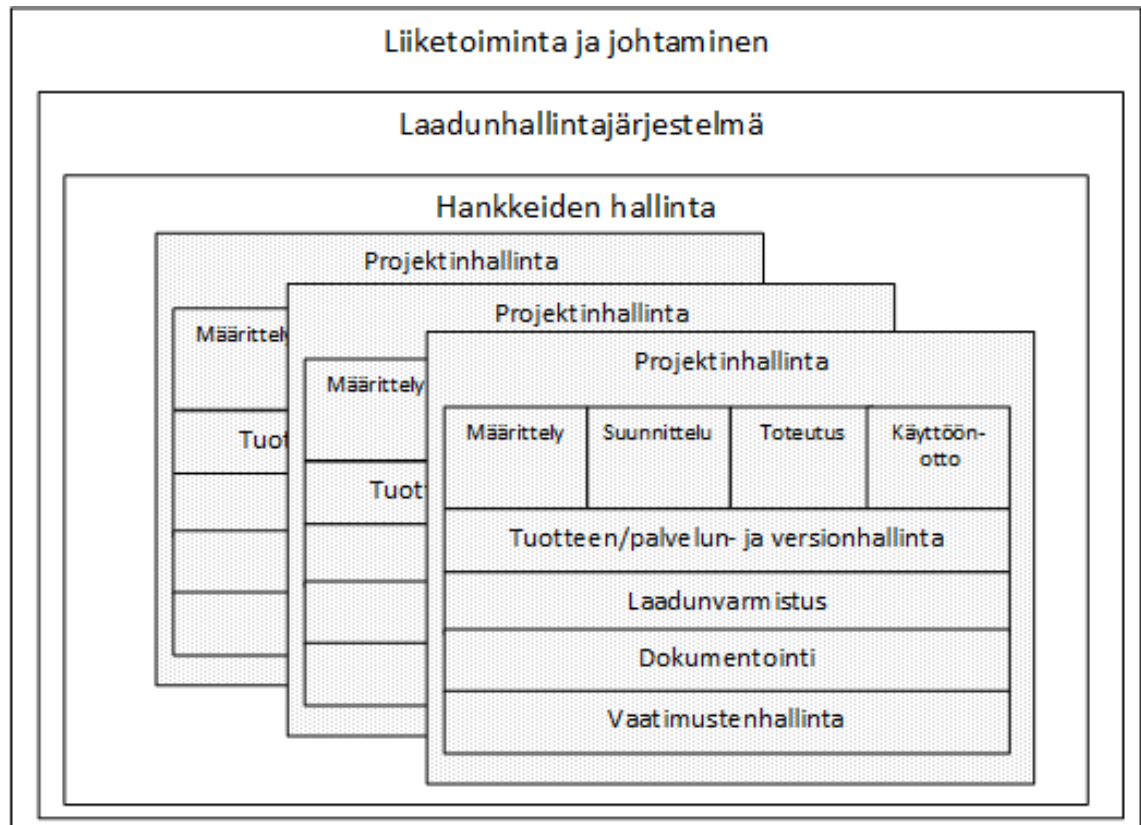
Tässä luvussa käydään läpi työhön liittyvä teoreettinen taustoitus. Teoriat ovat kerätty järjestykseen, jossa aloitetaan kuvaamalla tietotekniikan palveluyrityksen liiketoimintaa sovelluskehityksen ja palvelutuotannon kannalta. Tämän jälkeen esitellään asiantuntijoiden yksilö- ja tiimityön rakenteita ja näkökulmia tiimityön johtamiseen. Lean-ajattelun ja ketteryyden arvojen sekä periaatteiden kautta esitellään vakiintuneet viitekehykset ja menetelmät tehtävienhallintaan.

### 2.1 Tietotekniikka-alan yrityksen ja asiakkaan liiketoiminta

Tietotekniikan palveluyrityksen asiakkaalle hankkeet ja projektit ovat keino saavuttaa jokin liiketoiminnallinen tavoite tai tila. Asiakasyrityksellä yksittäiset projektit saattavat liittyä isompaan hankekokonaisuuteen, jossa yksittäinen projekti edustaa mitattavaa alun, sisällön ja lopun tilaa, jolla esimerkiksi sovellus- tai järjestelmäratkaisu saadaan vaatimustenmukaisesti toimintaan, sekä tukemaan hankkeen isompaa kokonaisuutta (Haikala & Mikkonen 2011).

Tietotekniikan palveluyritys vastaa täten sisäisen tai ulkoisen asiakkaansa tarpeeseen ohjelmistoratkaisuilla, jotka voidaan kehittää itse tai räätälöidä yrityksen ulkopuolista ohjelmistoa. Ohjelmistoratkaisuun liittyy myös olennaisena osana usein palvelutuotanto, jossa ohjelmiston lisäksi tarjotaan ohjelmiston ylläpitoon liittyvät palvelut. Tällöin tietotekniikan palveluyritys pyrkii tarjoamaan koko ohjelmiston elinkaaren palvelun eikä pelkkää käyttöönotto- tai ohjelmistokehitysprojektia.

Ohjelmistotuotannon toimittamisen organisoituminen tietotekniikan palveluyrityksessä voidaan jakaa isossa kuvassa itse liiketoimintaan ja sen ympärille muodostettuun laadunhallintajärjestelmään kuten mukaillen Haikala & Mikkonen (2011) esittää kohdassa Kuva 1. Laadunhallintajärjestelmä määrittelee tuotannon prosessit, esimerkiksi ohjelmistokehityksen ja niihin liittyvän palvelutuotannon. Hankkeiden hallinnan alla on itse asiakasprojektit, joissa ohjelmistokehitystä ja ohjelmistojen palvelutuotantoa suoritetaan.



**Kuva 1** Karkea yleiskuva esimerkkinä miten tietotekniikan palvelutuotantoa harjoittava yritys organisoii ohjelmistokehityksen ja palvelutuotannon (mukaillen Haikala & Mikkonen 2011, s 29)

Tietotekniikan palveluyrityksen asiakaskunnan tarpeiden mittakaava voi olla hyvin laaja riippuen millaista ja miten laajaa ratkaisua tarvitaan. Uusille vasta markkinaansa tai käyttökohdettaan hakeville sovelluksille voi olla tarpeen päästä nopeasti ja ketterästi liikkeelle. Tällöin on tärkeää kyetä tarjoamaan asiakkaalle mahdollisimman nopeasti nähtävä ja koettava ratkaisu. Tämä ratkaisu voi olla aluksi ns. demo- tai pienin toimiva tuote (MVP) osaratkaisu, joka esittelee lopullista mahdollista kokonaisratkaisua. Vastaavasti elinkaaren loppupuolen sovellusten tai laajojen monimutkaisten järjestelmäprojektien kysessä ollen, tarvitaan ratkaisujen jatkokehitykseen, ylläpitoon tai ajansaattamiseen hidasliikkeisempää lähestymistapaa. Asiakkaalla voi olla myös hallussaan ohjelmistoratkaisuja, joiden ylläpidon se haluaa ulkoistaa palvelutoimittajalle jona tietotekniikan palveluyritys voi toimia.

Tietotekniikan palveluyrityksen tulee siis organisoida tuottamaan näitä ratkaisuja löytämällä oikeat teknologiat, osaavat asiantuntijat ja luomalla tehokkaat tuotannon prosessit. Seuraavissa alaluvuissa käsittelemme aluksi kahta erilaista tietotekniikan palveluyrityksen palvelutyyppiä: *ohjelmistoratkaisun kehitystä ja ohjelmistoja palvelutuotantona.*

### 2.1.1 Ohjelmistoratkaisun kehitys

Ohjelmistoratkaisun kehittämisessä yksi tai useampi henkilö organisoituu tuottamaan valitulla teknologialla halutun sovelluksen annettujen vaatimusten mukaan. Tietotekniikan palveluyrityksessä on tyypillisesti useita ohjelmistokehityksen tiimejä, jotka ovat järjestetty tietyn osaamisen tai tekemisen ympärille. Ohjelmistokehittäjillä on tietoa ja taitoa koostaa eri teknologioilla haluttu ohjelmistotekninen ratkaisu (Haikala & Mikkonen 2011).

Ohjelmistotuotannon liiketoimintamallin näkökulmasta ohjelmistoa voidaan tehdä tuotekehityksenä, jolloin yritys itse arvio tuotteelle markkinan ja siihen tehtävät panostukset sekä vaatimusten kehityssuunnan. Tietotekniikan palveluyritys voi myös tuottaa ohjelmistokehityspalvelua, jossa kehitettävän tuotteen tai ratkaisun ohjaus tulee kokonaan tai osittain asiakkaalta. Ohjelmistokehittämisen hallinnointi ja organisoituminen riippuvat tällöin kulloinkin yrityksen valitsemasta liiketoimintamallista (Haikala & Mikkonen 2011).

Ohjelmistokehityksessä pyritään siis tuottamaan asiakkaalle ohjelmistoratkaisu, joka täyttää asiakkaan tarpeet ja luo asiakkaalle arvoa. Ratkaisu käsitteenä tietojärjestelmien monimutkaistessa sisältääkin käytännössä useita eri ohjelmistoja sekä niiden keskenään integroitua yhteistoimintaa. Ohjelmistoarkkitehtuurin eri tasot pyrkivät huolehtimaan ratkaisun teknologisesta eheydestä ja sen sopivuudesta asiakasyrityksen liiketoimintaprosesseihin. Ohjelmistoarkkitehtuurin hallinta voi kuulua tarjottavaan palveluun riippuen yrityksen valitsemista liiketoimintamalleista.

Kehitetyn ohjelmistoratkaisun tuotantoon saattamiseksi ja ylläpitämiseksi taas tarvitaan palvelutuotantoa, jonka järjestäminen tietotekniikan palveluyrityksessä on kuvattu seuraavassa alaluvussa.

### 2.1.2 Ohjelmistoratkaisun palvelutuotanto

Tietotekniikan palveluyrityksen asiakkaalleen tuottama palvelu alkaa palvelu- tai projektisopimuksesta. Sopimuksessa määritellään palvelun sisällön vaatimukset, mittarit ja palvelutaso. Palvelun sisältöön tyypillisesti kuuluu yhden tai useamman sovelluskokonaisuuden kehitys ja ylläpitäminen. Palvelussa asiakkaalle taataan sovittu sovelluksen saatavuus, hallitaan sovelluksen toiminnan muutokset ja tuotetaan asiakkaalle tukipalvelu. Palvelu päättyy sopimuksen rauettua. Palvelun sisällön sovellukset voivat olla edellisen alaluvun kuvauksen kaltaisia palveluyrityksen itsensä kehittämiä sovelluksia, asiakkaan omia sovelluksia, tai valmiita kolmannen osapuolen sovelluksia, jotka halutaan palvelun sisältöön ylläpidettäväksi.

Tietotekniikan palveluyrityksessä palvelutuotannon malli pohjautuu käytetyimmän ITIL-kokoelmaan (engl. Information Technology Infrastructure Library), joka on kokoelma

testattuja aktiviteetteja, parhaita käytäntöjä ja prosesseja. Nämä ovat otettu käyttöön useissa tietoteknistä palvelua tarjoavissa organisaatioissa, joten ITIL onkin laajimmin tunnistettu ja luotettu parhaiden käytäntöjen ohjekokoelma tietotekniikan palvelunhallinnassa tänä päivänä (ITIL 2011; Steinberg 2011).

ITIL määrittelee (tietoteknisen) palvelunhallinnan mallin (engl. service management) joukoksi organisaation kyvykkyyksiä, jotka järjestetään tuottamaan arvoa asiakkaalle. ITIL tarjoaa tuotettavan palvelun elinkaaren jokaiseen vaiheeseen parhaita käytäntöjä, joita yrityksen organisaatio voi soveltaa palvelutuotannossaan. ITIL jakautuu seuraaviin viiteen elinkaaren vaiheisiin (ITIL 2011; Steinberg 2011):

*Palvelustrategian* avulla asemoidaan palvelutuottaja oikeiden suunnitelmien ja mallien avulla saavuttamaan liiketoiminnan tuloksensa. Palvelustrategian myötä ymmärretään mitä kyvykkyyksiä organisaatiosta tarvitaan ja luodaan selkeä palvelutuotantomalli, jossa määritellään prosessit palvelutuotantoon.

*Palvelusuunnittelussa* strategiset tavoitteet muutetaan konkreettiseksi palveluportfolioksi ja palveluomaisuudeksi. Organisaatioon luodaan palvelunhallinnan suunnittelukyvykkyys, jolla varmistetaan palvelutavoitteiden saavuttaminen sekä standardien ja säännösten noudattaminen. Palvelusuunnittelu keskeisenä kattaa kapasiteetin-, palvelutaso-, saatavuuden- ja palvelun jatkuvuudenhallinnan.

*Palvelutransitiossa* suunnitellaan uusien, muuttuvien ja poistuvien palvelujen hallintaa. Vaiheessa huomioidaan riskit, hallitaan tuotantoon viennit, pidetään kirjaa palveluomaisuudesta ja ylläpidetään tietämyksenhallintaa.

*Palvelutuotannossa* varmistetaan, että palvelu on käytettävissä mahdollisimman tietoturvalisest ja häiriöttömästi. Palvelutuotannon tärkeimmistä osuuksista on häiriönhallinta, jonka tavoitteena on palauttaa normaali palvelutuotannon taso mahdollisimman nopeasti. Muita palvelutuotantovaiheen prosesseja ovat ongelman-, herätteen- ja pääsynhallinta.

*Jatkuva palvelun parantaminen* tarkastelee muuttuvia yrityksen liiketoimintaprosesseja ja pyrkii ohjaamaan näiden myötä parannuksien käyttöä tuotettavassa palvelussa.

ITIL-viitekehyksessä palvelutuotannon organisoitumisen funktioita on esimerkiksi palvelupiste (engl. service desk), jossa on nimetyt asiantuntijat, jotka reagoivat häiriöihin tai herätteisiin. Palvelupiste tarjoaa keskitetyn yhteydenottopisteen asiakkaiden palvelukokemuksen parantamiseksi (Steinberg 2011).

Käytännössä tämä voi tarkoittaa palveluyrityksen organisaatiosta riippuen hyvin monenlaista tapaa järjestyä yhden tai useamman asiakkaan palvelun hallitsemiseksi. Organisaati-

tiossa voi olla keskitetty joukko asiantuntijoita palvelutuotannon ja palvelupisteen tuottamiseen. Näitä joukkoja voi olla useampi asiakaskohtaisesti tai palvelua tuottava yritys voi integroitua osaksi asiakkaan omia palvelutuotannon prosesseja.

Toteutustavasta riippumatta, sovellusten ylläpitotehtävät tuottaa lopulta asiantuntijajoukko, jotka vastaavat asiakkaan tuki- ja palvelupyyntöihin sekä vastaavat sovellustuotannon hyvinvoinnista sovitulla palvelutasolla. Asiantuntijoiden täytyy tuntea tuettavat palvelut riittävällä teknisellä tasolla, jotta he pystyvät korjaamaan asiakkaan tai teknisen valvonnan tuottamat häiriöilmoitukset vioista ja täten palauttamaan tuotannon tilan normaaliksi.

## 2.2 Asiantuntija- ja tietotyön luonne

Tietotekniikan palveluyrityksen ohjelmistotuotannon mahdollistamisessa on useita työrooleja asiantuntijatasosta aina ylempään johtoon. Kirjallisuudesta löytyy useita tietotyön määritelmiä, joissa kuitenkin tyypillisesti korostuu tietotekniikan käyttö, työn edellyttämä suunnittelu ja näiden yhdistäminen. Tietotyössä myös teknologian lisäksi työprosessiin voi kuulua vaihteita, jossa luovuus korostuu yli rutiinien (Blom et al. 2001).

Tietotekniikan palveluyrityksen asiantuntijatyön työkenttään kuuluu tyypillisesti teknistä suunnittelua- ja kehitystyötä, jossa valituilla teknologioilla tuotetaan ratkaisua työtehtävän ongelmaan. Esimerkiksi ohjelmistokehittäjän ja järjestelmäylläpitäjän työrooleja voi-kin täten luonnehtia hyvinkin tietointensiiviseksi työksi, eli lyhyemmin tietotyöksi.

Tietotyötä luonnehditaan myös itseohjautuvaksi asiantuntijatyöksi (Blom et al. 2001). Tämä tarkoittaa, että tietotyötä tekevällä asiantuntijalla on tyypillisestä paljon valtaa omaan työnsä organisoimiseen ja suorittamiseen. Tästä voidaan johtaa, että yksilö tulee tarvitsemaan osaamista ja työkaluja oman työnsä johtamiseen. Täten korostuukin kuormituksen hallintaa tukeva johtaminen oikealla esimiestyöllä yksilön vastuun lisäksi. Tarvitaan johtajuutta, jolla voidaan löytää sopiva kuormittavuuden taso. Yksilötasolla tarvitaan riittäviä haasteita löytämään motivoiva ja tehokas suoritustaso, mutta terveydelle haitallista kuormitusta ei voida sallia (Kantolahti & Tikander 2010).

Psykososiaalisesta perspektiivistä työn kuormittavuuteen vaikuttaa monta tekijää. Kuormituskestävyyttä lisää monipuoliset työtehtävät, vaikutusmahdollisuus omaan työhön, jossa keskeisenä työmäärä, työtahti, työmenetelmät, työajat, työnjako ja työn läheiset henkilösuhteet. Tarvittava tuki työtehtävien suorittamiseen ja osallistuttaminen työyhteisön päätöksentekoon. Näiden ydinasioiden ollessa kunnossa, on parhaimmat edellytykset kestää kuormitusta. Jatkuvana keinona on myös tärkeää arvioida kuormitustilannetta osana työkulttuuria, kahvipöytä keskusteluna ja osana johtamista (Kantolahti & Tikander 2010).



## 2.3 Asiantuntijoiden yksilö- ja tiimityö

Digitalisaation tuomien teknologioiden kerrostumien laajentuessa, sekä tietotekniikan uppoutuessa yhä enemmän eri liiketoimintaprosesseihin, tarvitaan asiantuntijoilta yhä enemmän laajempaa ja syvempää osaamista. Ohjelmistotuotannon tekijöiltä tarvitaan osaamista asiakkaan liiketoiminnan ymmärrykseen, palvelutuotantoon ja itse teknologi-oihin, joilla tietotekninen ratkaisu asiakkaalle toteutetaan.

1900-luvun lopulta lähtien organisaatioiden rakenteet ovat muuttaneet muotoaan merkittävästi ja yksi merkittävimmistä muutoksista on työn organisoinnin muutos yksilötyöstä kohti tiimipohjaisuutta. Organisaatiot muuttuvat yhä enemmän tiimipohjaiseksi, koska niiden pitää vastata muutoksiin nopeammin, joustavammin ja mukautuvammin. Organisaatioon kohdistuu painetta olla luovampi, kekseliäämpi ja innovoivampi. Organisoituminen tiimeiksi yksilötyön sijaan mahdollistaa nämä ominaisuudet, koska organisaatioon tarvitaan yhä enemmän yksilötasoa laajempaa kyvykkyyttä (Kozlowski & Bradford 2013).

### 2.3.1 Mikä on tiimi?

Organisoitumiseen liittyvät kirjoitukset erottelevat työryhmän ja työtiimin. Tässä työssä puhutaan jatkossa vain tiimistä, jolle Kozlowski & Bradford (2013, s. 415) tutkimuskatselmuksessaan on koostanut kummallekin nimitykselle yhteiset piirteet:

- Jäsenenä kaksi tai useampaa yksilöä
- Jäsenet ovat järjestäytyneet suorittamaan organisaatiolle olennaisia tehtäviä
- Jakavat yhden tai useamman yhteisen päämäärän
- Jäsenten tekemisessä yhdistäviä asioita (esimerkkinä työnkulku, tavoite, tietämys, lopputulokset)
- Kommunikoivat sosiaalisesti keskenään (kasvokkain tai virtuaalisesti)
- Ylläpitävät ja hallitsevat tiiminsä rajojen olemassaoloa
- Ovat osa organisaation kontekstia, joka tuo tiimille rajat ja toiminnan rajoitteet, sekä ovat lisäksi yhteisvaikutuksessa muiden organisaation osien kanssa

Tiimiä voidaan tarkastella osana organisaatiossa olevaa systeemiä, jossa tiimi itsessään muodostuu oman näköisekseen. Laajemmat organisaatiokohtaiset piirteet kuten teknologia, rakenteet, johtaminen, kulttuuri ja ilmapiiri vaikuttavat aina tiimiin. Tällöin sanotaan niiden valuvan organisaatiosta ylhäältä-alas kohti tiimiä. Kuitenkin toisaalta tiimit aina edustavat approksimaatioita tiimin muodostavista yksilöistä, joiden yksilölliset taidot, tiedot ja piirteet muodostavat yhdessä käytännön toiminnan. Tiimin olomuodon voi siis sanoa olevan yhteismuodostuma ylhäältä-alas ja alhaalta-ylös vaikuttavista tekijöistä. Tiimi muodostuu, kypsyy ja kehittyy ajan kuluessa, joka kehittää tiimin keskinäistä dynamiikkaa. Tiimin rakenteeseen ja toiminnan ilmiöihin vaikuttaa aina aika. Ajan kuluessa

toiminnan ilmiöt: kuten ajatusmallit, ryhmäajattelu, tehokkuus, suorituskyyky alkavat kehittyä yksilötasolta tiimitasoiseksi kyvykkyydeksi (Kozlowski & Bradford 2013).

### 2.3.2 Tiimin muodostuminen ja kokoonpano

Tiimin käytännön toiminnan ilmiöt heijastelevat jäsenten eroavaisuuksia, kyvykkyyksiä, taitoja, demograafista taustaa ja persoonallisia piirteitä. Tiimi rakentuu monista sosiaalisista ja psykologisista prosesseista, eli yksilöt sosiaisoituvat tiimiin ja luovat sille yhteisen kontekstin (Kozlowski & Bradford 2013).

Optimaalista tiimikokoa eli jäsenten määrää ei ole kyetty yleistapauksena määrittelemään, vaan kirjallisuudessa vaihtelee useita eri määriä muutamasta aina yli tusinaan. Kozlowski & Bradford (2013, s. 419) kiteyttää seuraavasti eri tutkimuksista tiimikoon määrittämisen haasteita: Pääsääntöisesti oikean tiimikoon valinta on riippuvainen tehtävistä ja vallitsevasta toimintaympäristöstä. Esimerkkinä suurella tiimillä voi olla pääsy isompaan resurssimäärään, kuten aikaan, energiaan, rahaan tai osaamiseen. Toisaalta tämä voi myös tuoda liikaa hidastetta, jos ympäristötekijät muuttuvat huonommaksi. Tiimikoon kasvattaminen optimaalisesta lisää myös koordinaatio-ongelmia, joka alkaa myöhemmin heijastua suorituskyykyyn, sekä laskee jäsenten motivaatioita, koska tiimin jäsenyys alkaa tuntua hajaantuneelta eli menettää merkitystään tiiminä. Optimaalisen tiimikoon löytäminen on siis kompleksinen ongelma, joka pitää löytää tehtävien luonteen myötä, sekä huomioiden tiimin sisältä ja ulkopuolelta tulevat tarpeet.

Tiimin monimuotoisuuteen eli homo- tai heterogeenisyyteen vaikuttaa henkilöiden pinnalliset ominaisuudet, kuten ihmisrotu, sukupuoli ja syvällisempiä ominaisuuksia, kuten taustatekijät ja persoonallisuus. Heterogeenisyydestä tiimin aikaansaatavuuteen (eng. effectiveness) on tutkimuksia, jossa se on nähty vaikuttavan positiivisesti, mutta myös negatiivisesti, sekä olevan täysin merkityksetön Kozlowski & Bradford (2013, s. 420). Watson et al. (1993) havaitsi, että homogeeniset ryhmät olivat alussa suorituskyykyisempiä verrattuna heterogeenisiin, mutta myöhemmin heterogeeniset suoriutuvat paremmin. Tiimin monimuotoisuudesta (diversiteetistä) ei siis ole yksiselitteisiä tutkimuksia, voidaanko sitä pitää haluttuna ja täten tavoiteltavana ominaisuutena.

Tiimin vaikuttavuuteen on havaittu monimuotoisuuden lisäksi vaikuttavan persoonallisuus ja kognitiivinen kyvykkyys, jotka ovat yksilötasoisia ominaisuuksia toisinkuin aiemmin mainittu monimuotoisuus, joka on tiimin ominaisuus. Tunnetuimmat (Tupes & Christal 1961) esittelemät persoonallisuuspsykologiassa tunnustetut viisi yksilön persoonallisuuden piirrettä ovat:

- *Neuroottisuuden* (eng. emotional stability) matalalla tasolla henkilö ei järky tunne-elämässään pienistä ympäröivistä muutoksissa. Vahvalla neuroottisuuden tasolla tunne-elämän epätasapainoisuus näkyy ja koetaan herkästi negatiivisina tunteina ympäröiviä muuttuvia asioita kohtaan.

- *Ekstroversion* (eng. extraversion) vahvalla tasolla tunnetaan myönteisyyttä kokea uusia asioita ja yksilö energisoituu sosiaalisista suhteista. Matalalla ekstroversion tasolla käsittelee asioita enemmän oman maailmankuvansa kautta ja menettää energiaa sosiaalisissa tilanteissa.
- *Avoimuuden* (eng. openness to experience) vahvalla tasolla koetaan sisäisesti tarvetta laajentaa älyllisesti ja tunnetasolla tietämystä sekä kokemuksia ympäröivistä asioista. Matalalla tasolla pidättäydytään tutuissa toimintatavoissa ja ympäristössä.
- *Sovinnollisuuden* (eng. agreeableness) vahvalla tasolla koetaan kykyä sopeutua ja ymmärtää muiden ihmisten tilannetta ja toimintatapaa. Matalalla tasolla koetaan vaikeutta ja epämiellyttävyyttä sopeutumisessa muiden toimintaan tai tilanteen ymmärtämiseen.
- *Tunnollisuus* (eng. conscientiousness) vahvalla tasolla on kyky ja halu toimia järjestelmällisesti hoidettavien asioiden suhteen. Matalalla tasolla vaikea noudattaa sovittua aikataulua ja tehtäviä.

Persoonallisuuspiirteiden vaikutusta tiimin suorituskyykyyn ja aikaansaataavuuteen ei ole tutkittu paljoa. Meta-analyysit Bell (2007) ja Prewett et al. (2009) löysivät, että piirteiden merkitys on aikaansaataavuuteen vähäinen. Vähäisiä riippuvuuksia löydettiin myös siitä, että sovinollisuudella, ekstroversiolla ja neuroottisuudella oli vahvempia riippuvuuksia tiimin käyttäytymiseen ja toimintaprosesseihin kuin tiimin tuloksiin. Tämä voi tarkoittaa, että tiimin prosessien parantumisella, saavutetaan myös suorituskyydyn parantumista. Tiimin jäsenten persoonallisuustyyppien rakenne vaikuttaa myös vahvemmin vain silloin, kun tehdään tiimissä tehtäviä, jotka vaativat vahvaa keskinäistä riippuvuutta tekijöiden suhteen.

Yksilöiden arvot ovat uskomuksia halutusta ja hyväksytystä toimintatavasta tietyssä tilanteessa. Bell (2007) on löytänyt näyttöä siitä, että tietyt arvot, kuten kollektivismi eli yhteisön korostaminen yksilöllisyyden sijaan, on tärkeä tekijä tiimin suorituskyykyssä. Lisäksi Barrick et al. (1998) on löytänyt näyttöä, että tiimit joissa on enemmän ekstroversiota ja vähemmän neuroottisuuden piirteitä eli ovat tunnepohjaltaan vakaita, pystyivät parempaan ryhmän kanssakäymiseen, joka teki tiimistä yhdenmukaisemman ja pysyvemmän.

Samoin Stewart & Barrick (2004) esittävät, että jos tiimissä on paljon ekstroverttejä yksilöitä, voi olla hyvä ottaa mukaan vähemmän ekstrovertti henkilö, tai jopa introvertti. Vaihtoehtoisesti jos tiimi on täysin introvertteihin pohjautuva, voi olla tärkeää ottaa mukaan ekstroverttejä. Näiden piirteiden yhdistäminen on tärkeää, jotta voidaan luoda sopiva seikoite halutuista piirteistä. On tärkeää ymmärtää mistä piirteistä tiimi nykytilaansa koostuu, jotta voidaan määritellä jäsenten lisäystä varten oikea tavoitetilä, ja valita tähän tähtääviä yksilöiden piirteitä. Edelleen on myös muistettava, että tiimin tekemät tehtävät vaikuttavat valintaan, tehdäänkö enemmän homo- vai heterogeeniselle tiimille sopivampia tehtäviä (Neuman & Wright 1999).

Tiimin kokoonpanossa on siis hyvä ymmärtää kaikki edellä mainitut tekijät ja niiden keskinäinen vaikutus. Tämä on oleellista erityisesti tiimiä rakennettaessa ja henkilövalintoja tehdessä uuteen tiimiin. Olemassa olevaa tiimiä käsitellessä, on hyvä ymmärtää tekijöiden vaikutus nykyhetkeen. Perinteisesti tiimitehtävään henkilöä valittaessa, on voitu valita parhaimmat tulokset saanut jossain tiettyssä testissä, joka mittaa tiettyä ominaisuusjoukkoa. Tutkimuskatsauksen mukaan Kozlowski & Bradford (2013) osoittavat, että järkevämpää olisikin tehdä valinta tiettyjen määritellyn persoonallisuuden ominaisuusjoukkojen perusteella. Esimerkiksi narsisminen persoona voi lisätä ryhmän prosessien ja tuotosten luovuutta, mutta näiden yksilöiden liiallinen lisääminen rupeaa vähentämään tiimin luovuutta (Goncalo et al. 2010).

### **2.3.3 Sosialisaatio tiimiin ja tiimielinkaaren muodostuminen**

Tiimi voi muodostua kokonaan uusista jäsenistä ilman yhteistä historiaa tai muuta jäseniä aiemmin yhteen liittävää kontekstia. Tyypillisesti asiantuntijaorganisaatioissa muodostettavan tiimin jäsenillä on jo jonkinlainen historia organisaatioon, sen osiin ja mahdollisesti toisiinkin jäseniin. Tällöin historian tapahtumilla, tavoilla ja ihmissuhteilla on vaikutus uuden tiimin kontekstiin. Tiimin kehittymisen ja uuden jäsenen kannalta sosialisaatio on tärkeä mekanismi sulauttaa uusi tekijä tiimiin. Tiimi saattaa olla uudelle henkilölle tärkein elementti, joka vaikuttaa eniten yksilön organisaation ja yrityskulttuurin sosialisaatioon. Tällöin tärkeäksi muodostuu erityisesti sosialisaatio tiimiin (Kozlowski & Bradford 2013).

Sosialisaatio on prosessi, jossa uusi jäsen tulee osaksi tiimin yhteisöä. Uusi jäsen horjuttaa aina vakiintunutta rakennetta, ja kohtaa samalla työhön liittyvät kontekstit. Uusi jäsen haluaa luontaisesti sopeutua joukkoon ja on valmis ottamaan käytäntöjä vastaan. Samalla hän haluaa silti sopeutuvan myös tarpeisiin, arvoihin ja kyvykkyyksiin (Kozlowski & Bradford 2013).

Tiimiin sosialisoituminen on siis yhteistä vaikuttamista vanhojen ja uusien jäsenten kesken, jossa uudet tulijat vähentävät epävarmuuttaan mukautumalla tiimin kontekstiin. Pitkän kehittymisen omaavat tiimit tai tiimit joissa henkilövaihtuminen on hidasta, voivat olla vaikeampia uuden tulokkaan sosialisoitumisen kannalta. Mukautumista voi edistää esimerkiksi vaikuttamalla rekrytoinneissa ja jäsenten valikoinnissa, hakemalla tiimikohdattaisia vaatimuksia ja kyvykkyyksiä. Usein organisaatiot ovat perustaneet opiskelu, induktio ja orientaatio-ohjelmia, jotta normit, tavoitteet ja arvot saadaan sosialisoitua uudelle tulokkaalle. Tästä huolimatta on kuitenkin näyttöä siitä, että näillä prosesseilla on vain tehokas siirtymävaikutus, ja varsinainen intensiivinen sosialisaatio tapahtuu tiimin jäsenenä (Kozlowski & Bradford 2013).

Varhaiset klassiset tiimin muodostumisen mallit, esimerkkinä tunnetuin Tuckman (1965), jossa tiimin elinkaaren mallinnukseen esiteltiin neljä vaihetta ajan kuluessa:

- *Muodostumisvaiheessa* (eng. forming) tiimin jäsenet alkavat toteuttaa ensimmäisiä tiimin tehtäviä, jolloin he alkavat luomaan keskinäisiä sosiaalisia rakenteita. He määrittelevät ryhmän tehtäviä ja luovat tavat suoriutua niistä.
- *Kuohuntavaiheessa* (eng. storming) tiimissä huomataan, että tehtävät eivät aina ole yksinkertaisia, ja aletaan väitellä millä toimenpiteillä asiat saadaan ratkaistua. Erilaisia kannanottoja ja mielipidejakaumia syntyy, jotka johtavat myöhemmin konflikteihin.
- *Sopimisvaiheessa* (eng. norming) tiimi vihdoinkin saavuttaa halutun vastuunsa ja keskinäisen luotettavuuden. Alkaa muodostua sääntöjä, rooleja ja henkilöille asemaa. Tässä vaiheessa myös emotionaaliset konfliktit vähenevät ja yhteistyö alkaa sujumaan paremmin kohti yhteistä tavoitetta.
- *Hyvin suorittavassa vaiheessa* (eng. performing) tiimi pystyy suoriutumaan ja välttämään ongelmat, sekä ratkaisemaan annetut tehtävät.

Yllä avattu klassinen Tuckmanin malli perustui kuitenkin ryhmiin, joilla ei ollut aiempaa historiaa keskenään, eikä täten keskinäistä laajempaa kontekstia. Klassisiin malleihin on myös pyritty myöhemmin täydentämään vielä viimeinen vaihe, joka kuvaa tiimin päättymistä, eli hyvästelyä ja eroamista. Klassisista malleista tiimin kehityksestä on tutkimuksissa edetty suuntaan, jossa prosessit ovat enemmänkin jatkuvia ja kehitykseen liittyy ulottuvuutena myös suorituskky ja mukautuvuus. Tiimin suorituskky ja mukautuvuus nähdään dynaamisena seurauksena jatkuvassa kehittymisprosessissa. Tiimin suorituskkyyn ja vaikuttavuuteen on myös tunnistettu vaikuttavan aika, eli kun tiimi lähestyy lopulta hajoamispistettään. Tämän takia monen tiimin elinkaarta käsittelevän kontekstin yhteydessä on hyvä tunnistaa myös viimeinen vaihe ja sen vaikutus (Kozlowski & Bradford 2013).

Tuckman (1965) ja muut siihen pohjautuvat klassiset mallit keskittyvät kehittämään henkilöiden välisiä suhteita, sekä edistämään tavoitteita kohti yhteistä päämäärää. Nämä mallit eivät vielä tarjoa näkyvyyttä esimerkiksi taitojen kehittämiseen organisaatiossa. Nykyorganisaation tiimeissä uudet jäsenet usein tulevat jo organisaation sisältö, joten kulttuurillinen tietämys ja konteksti vähentävät itsessään jo sosiaalista epävarmuutta. Edellisessä alaluvussa todettiin, että tiimillä on useita rakenteellisia ominaisuuksia vaikuttamassa eikä tiimi tosiasiallisesti mene vaiheesta toiseen vaan siirtymät ovat enemmänkin epädiskreettejä.

Tiimit ovat myös lähes aina linkittyneet ulkoisiin sidosryhmiin ja konteksteihin, jotka määrittelevät paljon työtahtia, -nopeutta sekä työn jaksotusta, eli sitä miten ja milloin tiimi aktiviteettejään suorittaa. Tällöin tulee huomiotavaksi myös tiimin mukautuvuus ja joustavuus ulkoisiin tekijöihin.

Digitalisaatio on myös mahdollistaja yhdistäessään yksilöt globaalisti muodostaen virtuaalitiimejä. Tiimityössä voi siis korostua myös hajautetun- ja etätöön ulottuvuudet jos

asiantuntijayrityksen tiimit tai sidosryhmät työskentelevät fyysisesti eri paikoissa. Tosin tällöin voidaan puhua jopa käsitteellisestä ristiriidasta tiimityön ja etätyö mahdollisuuden välillä, koska mikään tekninen käyttöliittymä ei vielä korvaa kasvokkain läsnäoloa (Blom et al. 2001). Edellä listattuja prosesseja peilaamalla voidaankin todeta, että tärkeiden prosessien toiminta vaarantuu tai ainakin muuttuu etätyössä, jonka takia kommunikation ja yhteistyön on hyvä painottaa enemmän huomiota hajautetun etätyön tiimimallissa.

### 2.3.4 Tiimin johtajuuden näkökulmia

Tutkimuskatsauksessa Kozlowski & Bradford (2013, s. 447-450) mukaan monet tiimin vaikuttavuuteen mallit korostavat tiimin johtamisen roolia tärkeänä. Viitatus tutkimuskatselmuksen avainkysymys johtamisen suhteen tiimeille on: *Mitä johtajien pitäisi tehdä parantaakseen tiimin vaikuttavuutta?* Johtajalla voi olla tiimissä toiminnallinen rooli, mutta parantaakseen tiimin vaikuttavuutta, voidaan tutkia vaikuttavuuden parantumista lisäämällä itseohjautuvuutta ja jaettua johtamista tiimissä. Katselmuksessa tätä tarkastellaan kahdella perspektiivillä: *Tiimin johtajan toiminnallinen rooli ja itseohjautuva tiimi ja jaettu johtajuus tiimissä.*

#### *Tiimin johtajan toiminnallinen rooli*

Tutkimuskatsaus tiivistää tiimin johtamiselle kaksi olennaista kategoriaa:

- Tiimin työprosessien kehittäminen
- Tiimin suorituskyvyn ja vaikuttavuuden hallinta

Tiimiä perustettaessa johtajan rooli on ensimmäisenä koota toimiva tiimi, jolloin haasteena on löytää yksilöistä yhteensopiva ja hyvin integroitava joukko. Olemassa olevien tiimien kohdalla haaste on seurata ja reagoida henkilöiden vaihtuvuuteen, jossa uusien henkilöiden kohdalla tulee taas hoitaa orientaatio ja motivointi tiimin jäseneksi. Johtajan rooli on saada luotua uuteen tilanteeseen mahdollisimman koherentti eli yhdenmukainen tiimi, jolloin jäsenet pystyvät itseohjautumaan työtehtäviä tehdessään. Tiimin toimiessa ja aktiviteettien ollessa käynnissä, johtamiseen kuuluu suorituskyvyn ja vaikuttavuuden metriikan kerääminen, tiimin ympäristön ja sidosryhmien muutosten seuranta ja näihin liittyvien ongelmien ratkaisu. Tiimin johtaja voi vaikuttaa jäseniin suoraan, tai luoda mekanisme, jolla menetelmien hallinnan kautta vaikutus välittyy myös yksilötasolle. Virtuaalitiimeissä, joissa jäsenet saattavat olla hajautettu fyysisesti laajalle alueelle, joudutaan johtamista ja itsenäistä päätöksentekoa jalkauttamaan jäsenille.

Tiimityö tuo aina ympäristön, jossa on yksilöllisiä haasteita. Jäsenten yksilötason kehittämisen lisäksi, pitää jäsenille opettaa tiimityön toimintatapoja, yhteistä suorituskyvyn seurantaa, virheistä toipumista, kuorman ja resurssien jakamista.

#### *Itseohjautuva tiimi ja jaettu johtajuus:*

Itseohjautuvan tiimin piirteiden kuvauksia löytyy monia, mutta esimerkiksi (Hackman 1986; Manz & Sims 1987) määrittelee, että tiimille annetaan kokonaisia työtehtäviä ja heillä on vahva autonomia eli itsepäättämisoikeus tiimin toimintatavoista ja miten hallita työtään. Tiimillä on perinteisen johtamisen sijaan vastuu itse jakaa ja ottaa työtehtäviä, vastata laadustaan ja ratkaista jäsenten välisiä ongelmia. Itseohjautuvalla tiimillä tyypillisesti on kuitenkin johtaja, esimerkiksi projektin tai linjaorganisaation kautta, mutta johtajan tehtävä on luoda tiimille itseohjautuvuuden puitteita. Useat tutkimukset nostavat itseohjautuvan tiimin tuovan etua tuottavuudella, laadulle ja työhyvinvoinnille (Cohen et al. 1994; Lawler 1986; Manz & Sims 1987).

On löydetty, että yleisin epäonnistumisen syy itseohjautuvalle tiimille on johtaja, joka on liian autokraattinen ja puuttuu tiimin tehtäviin liiaksi, josta seuraa ettei tiimiin muodostu autonomian kokemusta (Stewart & Manz 1995). Myös vahva keskinäinen luottamus ja korkean tason itsenäinen autonomia johti suorituskyvyn laskuun, koska tiimin toimintaa ei enää tule seurattua tarkkaan (Langfred 2004).

Henrik Kniberg kirjassaan tiivistää osuvasti edellä mainitun teorian käytännön kokemukseen asiantuntijaorganisaation muutoksesta (Kniberg & Skarin 2010, s. 195): Tiimien muodostuessa yksilöistä kohti itseohjautuvia tiimejä, muodostuu tästä johtajille täysin uusia haasteita. Johtajien täytyy oppia ratkaisemaan henkilökonflikteja, määritellä jaettuja tavoitteita, käsitellä valituksia ja neuvotella sopimuksia. Tämä ei ole kivuton muutos johtamiselle, koska se tulee vaatimaan taitoja ja kuluttamaan energiaa.

### **2.3.5 Motivaation näkökulmia**

Motivaatioteoriat ja tutkimukset keskittyvät sosiaalipsykologian alueelle ja erityisesti yksilötasolle. Tutkimuskoosteessaan (Kozlowski & Bradford 2013, s. 450-452) tiivistää, että tiimiin kohdistuvan motivaation vaikutusta on tutkittu suhteellisen vähän ja enimmäkseen tutkimuksissa yksilötason toimintaa ekstrapoloidaan suoraan tiimitasolle. Myös Kilad & Kanfer (2006) toteaa, että tutkimukset ovat painottuneet yksilötasolle unohtaen tiimin sisäisten prosessien vaikutuksen, tai tiimitasolle unohtaen yksilöiden vaikutuksen.

Joidenkin tutkimusten mukaan tuottavuudelle voi tuoda jopa haittaa yksilötyön yhdistys isommaksi kokonaisuudeksi. Tällöin saattaa alkaa yleistymään ns. vapaamatkustaja -ilmiö (engl. social loafing), jossa yksilön suorittaminen laskee. Toisaalta on myös löydetty tekijöitä, jotka määrittelevät ilmiölle vastavoiman, kuten yksilöiden työskentely läheisten ystävien kanssa. Tällöin yksilö haluaa pitää omakuvansa positiivisena yhteisöön nähden ja yksilön osuus työstä tulee nähdyksi. Näillä tekijöillä mainitaan olevan myös suora riippuvuus tiimin suorituskyvyn parantumiseen (Zajong 1965; Bond & Titus 1983).

Useilla metatutkimuksilla on todettu, että tiimitason motivaatiotekijöistä merkittävimpänä on nähty se, että tiimillä on päämäärä, joka yleistyy tiimitason tavoitteeksi (O'Leary-Kelly et al. 1994; Kleingeld et al. 2011).

Tuottavuuden heikkenemistä voi Shepperd (1993) mukaan esiintyä tiimimotivaatiossa sen takia, jos jokin näistä kolmesta asiasta toteutuu yksilöillä:

- Ei koeta arvontuottoa omasta tekemisestään
- Koetaan epävarmuutta tekemisestä ja saavuttaa haluttu tavoite
- Koetaan oman tarvittavan panoksen olevan liian resursseja vaativia

Tiimin elinkelpoisuuden jatkuvuuteen ja innovointiin West & Anderson (1996) toi esiin neljä tekijää tiimille: *yhteinen visio, turvallisuuden tunne tekemisessä, tehtäväorientoitunut toiminta ja tuki uuden innovoinnille*. Tämän lisäksi organisaation on myös hyvä ymmärtää, että ryhmä käyttää energian ja jäsenten kyvyt hyvin sen sijaan, että hukkaisi tai jättäisi niitä soveltamatta. Lisäksi on hyvä löytää minkälaiset ryhmän keskinäisen kanssakäymisen kaavat toimivat, kehittyvät ja laajentuvat ajan kanssa, jotka lopulta johtavat suorituskyvyn parantumiseen.

Hackman (1987) nostaa tärkeänä esille, että tiimille tarjotaan avustusta koko elinkaaren ajan. Tiimille pitää tarjota uusia näkökulmia, jotka mahdollistaa suorituskyvyn ja suorittamisen nykytilan arvioinnin. Tämän lisäksi prosessien ja menetelmien avustusta pitää tarjota, jotta tiimi löytää todelliset ryhmäsynergiat. Esimerkiksi henkilöiden vaihtuvuus kytketään luonnolliseksi osaksi tiimin elinkaarta, jolloin tiimille vaihtuvuus ei tuo epävakautta ja päästään hyötymään uuden jäsenen tuomasta ajattelusta ja näkökulmista. Hackman korostaa myös elinkelpoisuuteen vaikuttamista jo panostamalla jäsenten valintaan.

### 2.3.6 Yhteenvetona tiimityö ohjelmisto- ja tietotekniikan palvelutuotannossa

Edellä läpikäytyjen alalukujen kautta voidaan todeta ohjelmistotuotannon asiantuntijatiimien luomisessa, johtamisessa ja toiminnassa olevan monta olennaista ja vaikuttavaa näkökulmaa.

Ohjelmistotuotannon prosessit ja ratkaistavat ongelmat ovat monimutkaisia, eikä useinkaan toistettavia kuten teollisissa tuotantoprosesseissa. Asiantuntijayrityksen resurssien eli *ihmisten* jaksaminen, oikea kuormitus, johtaminen ja motivointi ovat avainasemassa toimivan ohjelmistotuotannon perustana.

## 2.4 Lean toimintastrategiana

Lean on korkealla abstraktiotasolla filosofia, johtamistapa tai toimintastrategia, joka korostaa virtaustehokkuutta. Korkealla tasolla organisaatio etsii ja asettaa arvonsa, määrittää periaatteet ja ottaa oikeat menetelmät arvojensa tukemiseen. Matalalla abstraktiotasolla Lean käsittää kirjallisuudessa useita menetelmiä, joita voidaan käyttää toiminnan parantamiseen. Jokainen kirjallisuus määrittelee Leanin omalla tavallaan, mutta keskeistä



Leanissä on toiminnan jatkuva parantaminen ymmärtämällä asiakkaalle tehtävän työn arvonmuodostusta. Yksittäisillä menetelmillä ei automaattisesti oteta milloinkaan Leaniä käyttöön tai muututa Lean-organisaatioksi, vaan tarkastelu tulee olla aina koko organisaation läpiluotaavaa ja edellyttää Lean-ajattelun kokonaisvaltaista ymmärrystä (Modig & Åhlström 2013).

Lean on terminä kirjallisuudessa esitelty ensimmäisen kerran laajasti Womack et al. (1991) kirjassa, jossa tutkittiin japanilaisen autonvalmistuksen menestystekijöitä. Lean pohjautuukin Toyotan tuotantojärjestelmään Toyota Production Systemiin (TPS), jonka huomattiin olevan huomattavasti muita toimialan järjestelmiä edellä. Aiemmin tuotannonohjaamisessa oli paljon keskitytty sarjatuotantoon ja mittakaavaedun saavuttamiseen, kun taas Toyotan järjestelmästä tunnistettiin tekijöitä, jotka optimoivat työn virtausta ja keskittyivät enemmän asiakastarpeen arvon tuottamiseen. Toyotan menetelmiä on nyt tutkittu laajasti ja aiheen ympäriltä on kirjoitettu useita Lean-kirjoja (Modig & Åhlström 2013).

Koska ei ole mitään yksiselitteistä yhden ainoan lähteen listaa mitkä ovat tai eivät ole Lean-menetelmiä, on tässä listattu Lean-kirjallisuudessa yhteydessä toistuvasti mainittavia käsitteitä ja menetelmiä lyhyesti kuvattuna kohdassa Taulukko 1.

***Taulukko 1 Lean-menetelmien tunnettuja käsitteitä ja niiden tarkoitus***

<b>Käsite</b>	<b>Selite</b>
<b>Hukan eliminointi</b>	Prosessin vaiheista tunnistetaan hukkaa, eli tekijöitä tai tapah-tumia, jotka eivät tuota lopputuotokselle (asiakkaalle) arvoa. Alkuperäiset seitsemän hukkaa: tarpeettomat kuljetukset, va-rasto, liike, odottelu ja viivästykset, ylituotanto, laatuvirheet, korjaukset (Hines & Rich 1997).
<b>Virtaustehokkuuden korostaminen</b>	Organisaatiot perinteisesti pyrkivät olemaan resurssitehok-kaita, eli optimoimaan resurssille, esimerkiksi henkilölle tai koneelle mahdollisimman paljon tehtäviä. Usein kuitenkin asiakas, eli valmistettavan tuotteen saaja tai palvelun kokija, ei hyödy resurssitehokkuudesta, jolloin on keskeistä tehostaa toimintaa työn tai palvelun virtaavuuteen (Modig & Åhlström 2013).
<b>Just-in-time, imuoh-jaus, työntöohjaus, kanban</b>	Tuotannon valmistusketjussa pyritään varmistamaan, että oi-keat asiat tehdään juuri oikeaan aikaan. Tunnetuin menetelmä on Kanban, jossa viestisignaaleilla, esimerkiksi korteilla, viestitään valmistuksen edelliseen vaiheeseen tulevasta tar-peesta. Viestikortit auttavat myös työvirtauksen visualisoin-

	tia. Signaalit liikkuvat käänteisessä järjestyksessä valmistukseen nähden, eli tarve valmistukseen viestitetään lopusta alkuun. Lisäksi tuotannon määrää eri vaiheissa rajoitetaan eri parametreilla. (Sendil Kumar & Panneerselvam 2007).
<b>Jatkuva parantaminen, kaizen</b>	Osana johtamista jalkautetaan kulttuuria, jossa kaikkea toimintaa pyritään määrämuotoisesti parantamaan. Nykytilaa ei ikinä koeta täydelliseksi (Torkkola 2015).
<b>Arvovirtakuvaukset (Value Stream Mapping)</b>	Mallinnetaan jokin prosessi ja sen aktiviteetit, ja tunnistetaan mitkä aktiviteetit tuottavat asiakkaalle välitöntä tai välillistä arvoa. Tukee hukan eliminointia ja lisää ymmärrystä prosessiin, esimerkiksi läpimeno- ja jaksoaikojen mittaukseen. Arvovirtakuvaus työkaluna auttaa keskittämään toiminta kohti parempaa arvonluontia asiakasta kohtaan (Hines & Rich 1997).
<b>PDCA tai PDSA-sykli, Demingin laatuympyrä</b>	Nelivaiheinen parannusmenetelmä, jossa käydään kiertäen seuraavia vaiheita läpi (Michal & Paliszkievicz 2015): <ul style="list-style-type: none"> <li>- Plan (P), Suunnitellaan nykyistä tai uutta prosessia</li> <li>- Do (D), Jalkautetaan edellinen suunnitelma tai muutos</li> <li>- Check (C)/Study (S), kerätään tieto ja tulokset tehdyn muutoksen toiminnasta</li> <li>- Act, Adjust (A), muokataan edellisen vaiheen opitun tiedon mukaan prosessia. Opitun tiedon perusteella siirrytään syklin alkuun, eli suunnitteluvaiheeseen.</li> </ul>
<b>Pullonkaulateoria, kapeikkoajettelu (eng. Theory of constraints)</b>	Systeemissä on aina yksi pullonkaula, joka määrittää kokonaisuuden virtausnopeuden. Eliminoimalla pullonkaula, tulee voimaan toinen pullonkaula (Goldratt 1984).

## 2.4.1 Lean tietotekniikan palvelu- ja asiantuntijatyössä

1990-luvun Lean-kirjallisuus lähti valmistamisen ja perinteisen tuotannon lähtökohdista, jossa keskityttiin optimoimaan tuotannon materiaalivirtojen tai ihmisten fyysistä liikkumista tuotannon äärellä. Niissä ei keskitytty vielä asiantuntijatyöhön ohjelmistotuotannosta puhumattakaan (Modig & Åhlström 2013).

Viime aikoina Lean ajattelua on alettu tutkimaan ja soveltamaan yhä enemmän asiantuntijatyössä. Edellinen alaluku esitteli yhden Leanin käsitteen, joka oli virtaustehokkuus. Siinä tavoitteena on parantaa töiden virtaustehokkuutta ja erityisesti tehdä se resurssitehokkuuden sijaan. Organisaatio voi olla keskittynyt resurssitehokkuuteen, vaikka prosessin tehokkuuden kannalta olisi järkevämpää keskittyä virtaustehokkuuteen. Tietotekniikan palveluyrityksessä resurssitehokkuus voi konkreettisesti tarkoittaa sitä, että mahdollisimman monelle asiantuntijaresurssille yritetään löytää laskutettavaa työtä. Yrityksen mittarienkin puolesta tämä saattaa näyttää jopa tavoiteltavalta tilalta, jos liiketoimintamalli perustuu siihen, että asiantuntijatyötä skaalaamalla saadaan lisää laskutusta asiantuntijoiden työstä (Modig & Åhlström 2013).

Modig esittääkin kirjassaan tehokkuusparadoksin, jossa huomion keskittäminen resurssien käytön maksimointiin, synnyttääkin uuden lisätöiden tarpeen, joka taas vaatii lisää resursseja, työmäärää ja panoksia. Lisätarpeita syntyy esimerkiksi pitkästä työtehtävien lisääntyneestä läpimenoajasta, varastoinnista ja uudelleenaloittamisesta. Modigin tehokkuusparadoksin ratkaisu on se, että keskittymällä virtaustehokkuuteen, voidaan eliminoida paljon asioita, joita seuraa heikosta virtaustehokkuudesta. Läpimenoaika, keskenkäiset työt ja uudelleen aloitettavien töiden määrää laskee, kun emme liiaksi korosta resurssien käyttöä vaan töiden sujuvaa virtausta. (Modig & Åhlström 2013).

Työtehtäviä tietotekniikan palveluyrityksessä asiantuntijatasolla ovat esimerkiksi erilaisen muutosten teko ohjelmistoon, joko konfiguroimalla, häiriökorjaamalla tai ohjelmistokehittämällä. Näiden työtehtävien virtaavuuden tunnusmerkkejä on esimerkiksi, että tekijä tiedostaa mistä kanavasta seuraavan työn saa, sekä useamman kanavan tapauksessa kanavien keskinäisen prioriteetin. Lisäksi tulee pystyä tunnistamaan kuinka kauan tehtävän tekemiseen menee aikaa sekä minne työ toimitetaan ja milloin. Vastaavasti töiden virratessa tehokkaasti, asiantuntijatiimin johto kykenee kertomaan virtaavien töiden määrän, työpyyntöjen päivittäisen määrän ja valmistumisasteen (Torkkola 2015).

Asiantuntijatyössä itse asiantuntija – eli ihminen - onkin tuotannon tärkein resurssi:

*”Asiantuntijatyössä ihminen on olennainen osa työprosessia. Ihmisen hyvinvoinnilla on suora yhteys työn virheettömyyteen ja kehittämiseen. Ihmisen kannalta laadukas työympäristö on siksi merkittävä tekijä organisaation tehokkuuden toteutumisessa”* (Torkkola 2015, kappale 1: Oivallus)

Työvirtauksen suurin häiriötekijä voi olla keskeytykset tai informaation hankkiminen työtehtävää varten. Asiantuntija saattaa tarvita muilta tietoa, joka aiheuttaa taas muille asiantuntijoille keskeytyksen. Tutkimuksien mukaan töiden keskeytykset voivat lisätä käsitteleyäikää 40% prosenttia ja vaatimaan tehtävään takaisin pääseminen 10-15 minuuttia. Asiantuntijatyölle Torkkola esittää kirjassaan kolme pahinta vihollista tehtävien virtaukselle: *vaihtelu, ylikuormitus ja hukka*. Seuraavaksi on avattu mitä ne tarkemmin Torkkolan näkemyksen mukaan voivat asiantuntijatyölle olla (Torkkola 2015):

Vaihtelua asiantuntijatyössä:

- Henkilöiden osaamiserot
- Sisäänpäin tulevan asiakastyön kysynnän vaihtelua
- Asiantuntijan oman organisaation vaatimien tehtävien tekeminen
- Ennustamattomien häiriöiden kiireellinen korjaus

Ylikuormitusta asiantuntijatyössä:

- Henkinen töistä johtuva kuormitus, josta seuraa sairauspoissaoloja, motivaation laskua, joka johtaa oppimisen vähentymiseen ja työn tekemisen tapojen parantamisen.

Leanin seitsemän hukkaa asiantuntijatyössä:

- *Ylituotannossa* tehdään tehtäviä liian paljon, liian aikaisin tai varmuuden vuoksi. Tehtävien turha mittaaminen tai palaverit. Väärä priorisointi, jossa tehdään kiireellisiä, muttei ei kuitenkaan tärkeitä asioita.
- *Varastot* muodostuvat asiantuntijan keskeneräisestä työstä, jota voi olla sähköpostit, projektin muiden työtä odottavat työtehtävät tai raportointi.
- *Odottamista* on kun työtehtävä seisoo jonossa odottamassa seuraavaa asiantuntijaa tai sen välittämistä eteenpäin
- *Ylimääräistä materiaalin liikettä* on esimerkiksi tiedon syöttäminen käsin toiseen järjestelmään tai useamman sovelluksen käyttäminen. Toimistotyössä sitä voi olla myös ylimääräinen kävely huonon tilasuunnittelun takia.
- Töiden *siirtäminen* henkilöltä tai osastolta toiselle. Jos työn suorittaminen vaatii asiantuntijoiden välistä tiivistä kommunikointia, heidät pitäisi sijoittaa mahdollisimman lähelle toisiaan tai mahdollistaa tiivis yhteydenpito.
- *Virheitä ja uudelleen tekemistä* ovat asiakkaan tyytymättömyys tai edellisen työvaiheen toimittama virheellinen työ tai tieto, jolloin uudelleenpalautuva työ syö uudelleen kapasiteettia. Virheet aiheuttavat kumuloituvaa vaihtelua työketjussa, joten alkupään virheiden torjunta on erityisen tärkeää.
- *Epätarkoituksen mukaista käsittelyä* on työ, jota asiakas ei oikeasti halua tehtäväksi. Työn tekeminen suurissa erissä sen sijaan, että työ olisi organisoitu pienempiin osiin tai tehtäisiin suuremmilla tietojärjestelmillä tai laitteilla, vaikka pienempi todellisuudessa riittäisi.

Tiimin suorituskyvyn kannalta paras mittari on työtehtävien virtausprosessin mittaus. Perinteiset tulostimet talouden tai varaston kierron suhteen kertovat enemmän toiminnasta. Esimerkiksi, jos mitataan häiriöhallinnan vastausaikoja ja tulos ylittää tavoiteajan, on tällöin kyse tulostimista. Siitä ei voi päätellä miksi vastaukset kesti yli tavoitteen tai miten prosessia pitäisi parantaa, mutta prosessin parannustarpeen se ilmaisee selvästi (Torkkola 2015).

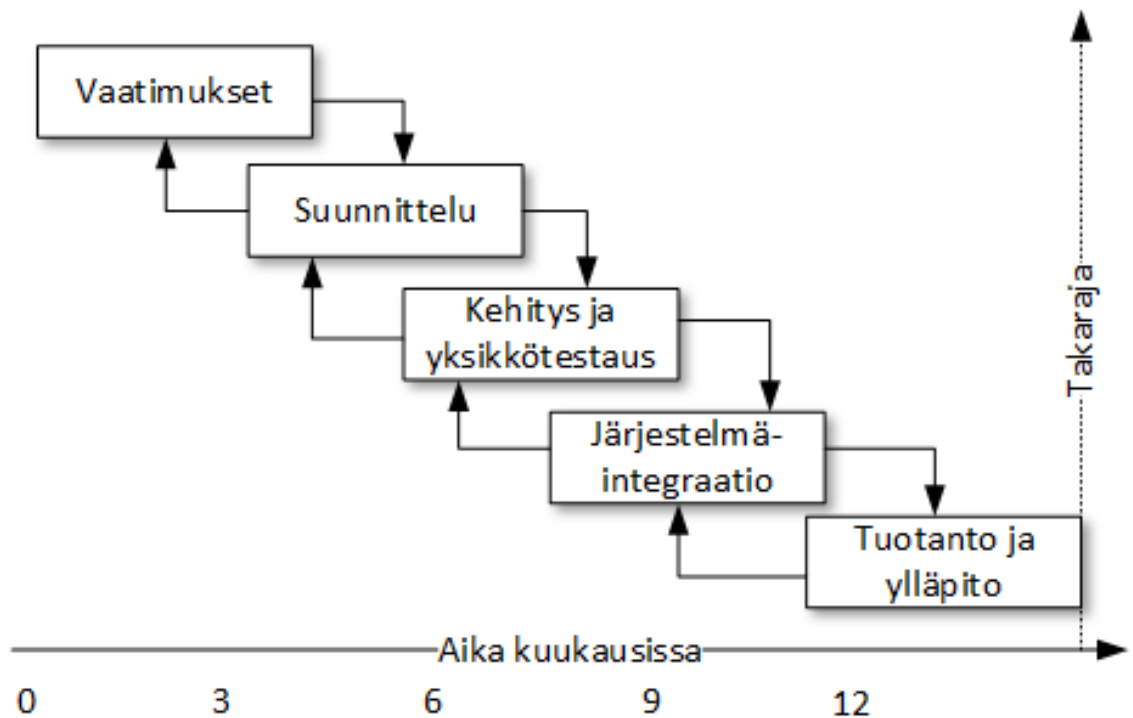
## 2.5 Perinteisistä suunnittelumalleista kohti ketteryttä

Tässä alaluvussa esitellään ensiksi historiallinen ohjelmistotuotannon vesiputousmalli, jonka jälkeen käsitellään ketteryyden käsitettä arvoina ja periaatteina sekä kuvataan, miten niillä vastataan vesiputousmallin ongelmiin.

### 2.5.1 Historiallinen vesiputousmalli ja sen ongelmat

Vasta vain muutama vuosikymmen on luotu ison mittakaavan ohjelmistoratkaisuja, joten myös suunnittelumallit ovat olleet murroksessa. Ensimmäiset suunnitelmalähtöiset ohjelmistokehitysmallit ovat olleet ohjelmistotuotannossa jo yli 30 vuotta. Kirjallisuudessa viitataan esimerkiksi Royce W. (1970) artikkeliin, josta vesiputousmallin eli vaiheistetun kehitysmallin vaikutteet olisi alun perin saanut alkunsa. Joskin siinäkin kuvattiin vain yksinkertaistettu tapaus, jota olisi tarkoitus käyttää kun sovellus on jo prototyypitetty ja asiakasvaatimukset tiedossa. Malli kuitenkin yleistyi yksinkertaisuuden vuoksi käyttöön, koska vaiheistus tuntui loogiselta ja selvemmältä. Vesiputousmalli oli kuitenkin edistystä ohjelmistokehityksen alkutaipaleen ”*koodaa-ja-kokeile*” -etenemiseen, joka oli ensimmäinen intuitiivinen tapa ennen määrämuotoista suunnittelun aikakautta (Leffingwell 2008).

Kuva 2 esittää vesiputousmallin perinteiset suunnittelun vaiheet vaatimusten suunnittelusta kohti valmiin sovelluksen tuotantoa ja ylläpitoa. Jokaisesta yksittäisestä vaiheesta voidaan palata taaksepäin, esimerkiksi uudelleen korjaamaan edellisen vaiheen tekemistä. Vesiputousmallin etuja oli looginen tapa löytää, analysoida ja suunnitella monimutkaisenkin järjestelmän vaatimukset etukäteen, jonka takia se on osittain yhä käytössä tänäkin päivänä.



**Kuva 2 Perinteisen vesiputousmallin vaiheet (mukaillen Leffingwell 2008, chapter 2 *Why the waterfall Model Doesn't work*)**

Ohjelmistotuotannon ongelmana vesiputousmallissa oli suunnittelun jälkeen tapahtunut myöhäinen tuotantoon vienti, jonka aikana käyttäjien vaatimukset olivat jo ehtineet muuttua. Lisäksi järjestelmäintegraation jättäminen viimeiseksi vaiheeksi ennen tuotantoonottoa, tuotti yllättäviä ongelmia, joka pidensi edelleen sovelluksen kehitysaikaa. Vesiputousmallin ongelmia on tutkittu paljon ja niistä on myös tilastollista näyttöä (Larman 2004). Keskeiset vesiputousmallin toteutumattomat tai vääriksi todetut neljä oletusta ja niiden vasta-argumentit olivat Leffingwell (2008) mukaan:

1. Hyvät määrittelyt ovat tuotettavissa, jos varataan aikaa niiden ymmärtämiseen
  - Ohjelmisto ei ole käsin kosketeltava käsityö, jota olemme historiassa tottuneet valmistamaan ja tunnistamaan työhön liittyvät riippuvuudet. Materiaalisen tuotteen toiminta on helppo ymmärtää ja nähdä konkreettisesti miten se vastaa tarpeita. Ohjelmisto on taas immateriaalinen tuote, jossa valmistetaan ratkaisu, jonka ajatellaan toimivan.
  - Kehitysprosessissa asiakkaat arvioivat toimiiko kehittäjien visio, ja samaan aikaan tarkentavat omaa visiotaan tuotteesta. Määrittelyssä yritämme kirjoittaa nämä muuttuvat elementit kirjalliseen vaatimusten muotoon, jota on vaikea lukea ja kirjoittaa. Vaatimukset pelkkänä pohjana ohjelmiston suunnitteluun eivät toimi.
  - Tuotettava ohjelmisto on myös osa jonkin organisaation liiketoiminnan prosesseja. Toimittamalla prosessien käyttöön uudistettu ohjelmisto, se

muuttaa aina myös liiketoimintaa, joka taas muodostaa uusia muuttuvia vaatimuksia.

2. *Vaatimukset muuttuvat kehitysprosessin aikana niin vähän, että koko suunnitelmaa ei tarvitse miettiä uudelleen*
  - Ohjelmistokehityksen aikana vaatimukset muuttuvat aina. Mikään liiketoiminta ei pysy staattisena vaan on enemmän tai vähemmän muutoksessa. Jos kehitys on nopeaa ja muutokset todella pieniä, voidaan tavoite saavuttaa, mutta muussa tapauksessa edessä on vaikeuksia.
3. *Järjestelmäintegrointi on tarpeellinen prosessi, ja voimme ennustaa miten se tulee menemään arkkitehtuurin ja hyvän suunnittelun pohjalta*
  - Ajateltiin, että huolellinen arkkitehtuurisuunnittelu voisi etukäteen näyttää kaikki tulevat järjestelmäintegraation ongelmat. Siitäkin huolimatta vaikka sovellusta kehittäisi useampi tiimi. Historiassa on yritetty tehdä jopa kokonaisia tiimejä keskittymään pelkkään järjestelmäintegraatioon, mutta silti epäonnistuttiin, koska ongelma oli liian kompleksinen. Muutoksia tapahtuu monessa eri tasossa. Esimerkiksi teknologiat muuttuvat kehityksen aikana ja oletukset integraatioista ovat usein väärässä ja löytyvät liian myöhään.
4. *Uuden ohjelmistoinnovaation tutkimus ja kehitys voidaan tehdä ennustettavassa aikataulussa ja onnistua siinä ensimmäisellä kerralla.*
  - Kirjoittajan kokemukseen perustuen tiimit voivat estimoida työmäärään jopa kaksinkertaisesti väärin vaikka käytössä olisi samat resurssit ja saman alueen uusi sovellus
  - Vesiputousmallin suunnittelu etukäteen on kompleksista, koska erilaiset kehitettävän systeemin osat siirtyvät eri vaiheisiin, joten tosiasiaissa meillä on useita rinnakkaisia vesiputouksia yhden systeemin sisällä.

Näistä seikoista johtuen vesiputousmallin kehitysvaiheen aikataulu siirtyy hyvin todennäköisesti kohti järjestelmäintegraatiovaihetta. Tästä taas seuraa se, että on paljon testejä vielä tekemättä, ja todennäköisesti on tunnistettu uusia rinnakkaisia vaatimuksia, joiden testejä on myös käynnissä. Lopulta ymmärretään, että järjestelmän integraatio ei toimi tai kokonaisuus ei muuten ole ennustetun mukainen. Tässä vaiheessa on käytetty jo paljon resursseja asioihin, jotka eivät vielä toimii, emmekä edes tiedä vielä oletuksista, jotka olivat väärin ja johtivat tähän tilanteeseen. Tästä johtuen todellinen valmistumisaika siirtyy eteenpäin. Valmistusajan siirtymisestä seuraa taustalla olevien liiketoimintavaatimusten muuttuminen, eikä alkuperäiset suunnittelun pohjalla olleet vaatimukset enää välttämättä päde (Leffingwell 2008).

## 2.5.2 Ketteryyden käsite, arvot ja periaatteet

Ketteryys tarkoittaa semanttisesti nopeutta vaihtaa suuntaa ja vastata muutoksiin. Ketteryyden korostaminen voi vieläkin tuntua muoti-ilmiöltä, mutta se alkoi muodostumaan jo

90-luvun lopulla kun tunnetut ohjelmistoalan mielipidevaikuttajat eri puolilta maailmaa alkoivat puhumaan yhä enemmän erilaisista menetelmistä, jotka kaikki lähestyivät samaa tavoitetta. Haluttiin onnistua luomaan toimiva ja luotettava sovellus, joka kehitetään nopeasti ilman hukkaa. Ketteryys ei rajoitu tekniseen mukautumiskykyyn kehittää ja julkaista ohjelmistoja nopeasti, vaan sillä on olennainen linkki asiakkaan liiketoiminnan tarpeisiin, jotka voivat muuttua nykypäivänä yhä tiheämmin. Ketteryyden ajuri tuleeikin halusta olla kyvykäs vastaamaan nopeasti muuttuviin tarpeisiin, jolla yritys tai yhteisö taistelee omasta kilpailukyvystään (Leffingwell 2008).

Ohjelmistotuotannon ketterä kehittäminen popularisoitui ja sai laajasti näkyvyyttä kun vuonna 2001 ohjelmistoalan mielipidevaikuttajat allekirjoittivat ja julkaisivat ketterän ohjelmistokehityksen julistuksen (eng. Agile Manifesto). Siinä he totesivat, että on parempi tapa luoda ohjelmistoja arvojen ja periaatteiden kautta. Julistus sisältää ketterän tekemisen neljä arvoa ja 12 periaatetta, joita monessa metodissa ja viitekehyksessä noudatetaan (Highsmith 2001, verkkosivuston käännös):

*Löydämme parempia tapoja tehdä ohjelmistokehitystä, kun teemme sitä itse ja autamme muita siinä. Kokemuksemme perusteella arvostamme:*

- *Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja*
- *Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota*
- *Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja*
- *Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa*

*Jälkimmäisilläkin asioilla on arvoa, mutta arvostamme ensiksi mainittuja enemmän.*

Julistuksen edellä mainitut arvot pohjautuvat periaatteisiin, joita julkaistiin 12 kappaletta:

- *Tärkein tavoitteemme on tyydyttää asiakas toimittamalla tämän tarpeet täyttäviä versioita ohjelmistosta aikaisessa vaiheessa ja säännöllisesti.*
- *Otamme vastaan muuttuvat vaatimukset myös kehityksen myöhäisessä vaiheessa. Ketterät menetelmät hyödyntävät muutosta asiakkaan kilpailukyvyyn edistämiseksi.*
- *Toimitamme versioita toimivasta ohjelmistosta säännöllisesti, parin viikon tai kuukauden välein, ja suosimme lyhyempää aikaväliä.*
- *Liiketoiminnan edustajien ja ohjelmistokehittäjien tulee työskennellä yhdessä päivittäin koko projektin ajan.*
- *Rakennamme projektit motivoituneiden yksilöiden ympärille. Annamme heille puitteet ja tuen, jonka he tarvitsevat ja luotamme siihen, että he saavat työn tehtyä.*
- *Tehokkain ja toimivin tapa tiedon välittämiseksi kehitystiimille ja tiimin jäsenten kesken on kasvokkain käytävä keskustelu.*
- *Toimiva ohjelmisto on edistymisen ensisijainen mittari.*



- *Ketterät menetelmät kannustavat kestävään toimintatapaan. Hankkeen omistajien, kehittäjien ja ohjelmiston käyttäjien tulisi pystyä ylläpitämään työtahtinsa hamaan tulevaisuuteen.*
- *Teknisen laadun ja ohjelmiston hyvän rakenteen jatkuva huomiointi edesauttaa ketteryyttä.*
- *Yksinkertaisuus - tekemättä jätettävän työn maksimointi - on oleellista.*
- *Parhaat arkkitehtuurit, vaatimukset ja suunnitelmat syntyvät itseorganisoituvissa tiimeissä.*
- *Tiimi tarkastelee säännöllisesti, kuinka parantaa tehokkuuttaan, ja mukauttaa toimintaansa sen mukaisesti.*

Julistuksen keskiössä oli *kyky toimittaa jatkuvalla jakelulla toimiva sovellus* niin, että kehitys tukee sen *nopeasti muuttuvia vaatimuksia*. Nämä ketterän ohjelmistokehityksen arvot ja periaatteet ovat keskeisiä kivijalkoja kun sovelletaan uusia menetelmiä ja kehitetään asiantuntijatiimin toimintaa (Leffingwell 2008).

### **2.5.3 Miten ketterät menetelmät vastaavat vesiputousmallin ongelmiin**

Ketterille menetelmille on yhteistä, että ne yrittävät välttää kaikkia vesiputousmallin ongelmakohtia. Alaluvun 2.5.1 lopussa esiteltiin vesiputousmallin ongelmia ja perusteltiin miksi sen kanssa usein epäonnistutaan. Leffingwell (2008) asemoi esitetyt ongelmakohdat seuraavasti ketterän kehityksen lähtöoletuksiksi:

- Emme oleta, että edes asiakas tai ylipäättään kukaan voisi täysin ymmärtää kaikkia vaatimuksia täysin etukäteen
- Emme oleta, että muutos on pientä ja hallittavaa. Oletamme sen sijaan, että muutos on jatkuvaa, jonka takia tuotamme sovellusta pienissä inkrementeissä hallitaksemme muutoksen
- Oletamme, että järjestelmäintegraatio on tärkeä vaihe ja vähentää riskiä. Alamme integroida heti alusta ja koko kehityksen ajan, jokaisella tuotettavalla inkrementillä.
- Ymmärrämme ja oletamme heti alusta, ettei ole mahdollista tehdä uutta innovatiivista sovellusta kiinteällä aikataululla ja toiminnollisuuksilla. Sen sijaan voimme tarjota tärkeitä ominaisuuksia ensiksi, jonka jälkeen tiiviillä asiakkaan palautteella saamme ymmärryksen onko ratkaisu toivottu.

Taulukko 2 kuvaa vesiputousmallin ja ketterän kehittämisen prosessien eroavaisuuksia Leffingwell (2008) mukaan. Erot perusteluineen kertovat, kuinka ketteryys vaikuttaa kaikkiin ohjelmistokehityksen prosessin vaiheisiin.

**Taulukko 2 Ohjelmistotuotannon eri prosessien erot siirryttäessä vesiputousmallista ketterään (Leffingwell 2008)**

Prosessi	Vesiputous mallissa	Ketterässä mallissa	Perustelu
Onnistuvuuden mittari	Suunnitelman noudattaminen	Nopea vaste muutokseen, kokoajan toimiva sovellus	Perinteisten projektin ositus muutetaan kohti arvoajattelua. Suunnitelma on joustava, sovellus kullakin hetkellä on paras tuotos vaatimuksista. Sovellus voidaan millä tahansa hetkellä julkaista.
Johtamiskulttuuri	Johda ja kontrolloi	Johtaminen oppimaan, yhteistyöhön ja itseohjautuvuuteen	Johto määrittelee suunnan. Johto ei anna kiinteää sisältöä, eikä aikataulua tai teknisiä resursseja. Tiimi päättää miten paljon työtä se ottaa sovitulla aikajänteellä. Tiimi itseohjautuu tarvittavasti saavuttaakseen tavoitteet. Johdon tehtävä on poistaa esteet ja luottaa tiimiin.
Vaatimukset ja suunnittelu	Suurta ja etukäteen	Jatkuvaa ja kokoai-kaista	Vaatimuksiin, arkkitehtuuriin ja suunnittelupäätöksiin käytetty aika pidetään mahdollisimman lyhyenä. Pyritään julkaisemaan nopeasti ja

			jatkuvasti arvoa tuottavia julkaisuja.
Kehitys ja toteutus	Ominaisuudet tehdään ja testataan rinnakkain	Tehdään ja yksikötestataan sarjassa	Löydetään tärkeimmät ja kiireellisemmät ominaisuudet, jotka priorisoidaan julkaistavaksi. Ohjelmoinnissa vain testattu, toimiva ja integroitu koodi on hyväksyttävä. Kommunikaatio on jatkuvaa ja kaikki tiimin jäsenet tietävät mitkä ovat toimituksen tavoitteet ja mitä heiltä odotetaan milloinkin.
Testaus ja laadunvarmistus	Suurta, suunniteltua ja myöhäistä	Jatkuvaa, yhtäaikaista ja ennakkoivaa	Testaus ei ole tuotannon elinkaaren vaihe tai organisaation yksikön vastuulla. Testaus on jatkuva osa kehitystä ja jakelua.
Suunnittelu ja aikataulutus	PERT, yksityiskohmainen, kiinteä sisältö, arvioitu aika ja resurssit	Jatkuva suunnittelu ja iterointi tuotannon edetessä systemaattisesti ja rutiniinomaisesti	Suunnittelu ei katoa ketterissä menetelmissä vaan päinvastoin, suunnittelu on intensiivisempää ja sitä tehdään kahdella tasolla. Ison tason suunnittelu julkaisuihin ja tarkemman tason suunnittelu julkaisuiteraatioihin.

Yhteenvetona ketteryys asettaa siis kokonaan uudet oletukset ohjelmistokehitys prosessista. Paradigma muuttuu lyhyesti kuvaten niin, että muututaan kohti nopeaa jakelua priorisoitujen vaatimusten pohjalta. Iso haaste on ymmärtää myös, että laajuuden ja aikataulun taistellessa, aikataulu voittaa aina (Leffingwell 2008).

## 2.6 Ohjelmistotuotannon viitekehyksiä ja menetelmiä

Tässä alaluvussa esitellään käytetyimmät ohjelmistotuotannon viitekehykset ja menetelmät, joilla tiimin kehittymistä, tehtäviä, rooleja ja työkuormaa voidaan hallita. Monet ketterät viitekehykset omaavat yhdenmukaisia piirteitä ja ominaisuuksia, mutta kaikilla on oma terminologiansa ja lähestymiskulmansa.

Tähän valitut esiteltävät menetelmät ja viitekehykset olivat käytetyimpiä VersionOne (2018) tekemän vuosittaisen kyselytutkimuksen mukaan. Kyselytutkimuksessa viitekehysistä Scrum, Scrumban ja yhdistetty Scrum ja eXtreme Programming muodostivat jopa 70% käytetyistä ketteristä menetelmistä kuluneella vuodella.

### 2.6.1 Scrum – ketterä viitekehys tuotekehitykseen

Scrum on tunnetuin ja käytetyin viitekehys tuotteen kehittämiseen erityisesti ketterässä ohjelmistotuotannossa (VersionOne 2018). Tämä alaluku kuvaa Scrum viitekehysten toiminnan mukaillen Scrumin kehittäjien julkaisemaa The Scrum Guidea käyttäen (Schwaber & Sutherland 2017). Scrum on viitekehys, jonka sisällä voi käyttää monia erilaisia tekniikoita ja hienojakoisempia prosesseja. Scrum antaa työkalun ensisijaisesti tuotekehityksen hallintaan ja työtapoihin, joilla voi jatkuvasti parantaa tuotetta, tiimiä ja työympäristöä.

Scrum-menetelmää on käytetty laajasti tuotekehityksessä. Käyttökohteita on laitteistosta ohjelmistoihin, hallintoon, markkinointiin ja tuotannonohjaukseen. Scrum on osoittautunut erityisen tehokkaaksi iteratiivisessa ja inkrementaalissa tuotekehityksessä ja tiedon siirrossa. Scrum pohjautuu empiriseen prosessien ohjausteoriaan. Empirismi tarkoittaa sitä, että tieto tulee kokemuksesta ja siitä mitä tiedetään kullakin hetkellä. Scrum tarjoaa iteratiivisen ja inkrementaalisen lähestymisen ennustettavuuden ja riskin kontrollointiin tuotekehitysprosessin edetessä.

Scrum-viitekehyksellä on kolme keskeistä koko toimintaa ohjaavaa peruspilaria:

- *Läpinäkyvyydessä* (engl. transparency) kehitysprosessi on näkyvä kaikille lopputulokseen liittyville sidosryhmille. Työstä ja tuloksesta vastaaville tavoitellaan yhteistä määrittelyä ja käsitystä, jotta saavutetaan mahdollisimman tarkka yhteinen kuva yhteisistä päämäärästä.
- *Tarkkailussa* (engl. inspection) viitekehysten käyttäjät tarkkailevat säännöllisesti artefakteja ja edistymistä

- *Sopeuttamisessa* (engl. adaptation) puututaan prosessiin, jos havaitaan heiluntaa yli hyväksyttyjen rajojen tai lopputulos ei ole hyväksyttävä. Säädetään prosessia tai materiaalia niin, että vaihtelu saadaan minimoitua.

Scrum-tiimi koostuu tuoteomistajasta (engl. Product Owner, PO), kehitystiimistä (engl. Development Team) ja Scrum Masterista (engl. Scrum Master). Scrum-tiimien sanotan olevan poikkitoiminnallisia. Itseohjautuvuus tiimillä tarkoittaa kyvykkyyttä valita parhaat tavat työskennellä sen sijaan, että niitä tarvitsi ohjata ulkoa. Poikkitoiminnallisuus tarkoittaa kykyä ratkaista tiimin sisällä kaikki eri teknologioita tai taitoja vaativat asiat. Tiimi tuottaa tuotteen iteratiivisesti, eli palasissa, jotka mahdollistavat sen hetken tärkeimmäksi prioriteetiksi nostetun toiminnallisuuden. Scrumin roolit lyhyesti esiteltynä:

**Tuoteomistajan** rooli on maksimoida kehitystiimin valmistaman tuotteen arvo. Tuoteomistaja on täysin yksin vastuussa tuotteen tehtävälisan (engl. Product Backlog) sisällystä ja priorisoinnista. Hän voi kuitenkin delegoida tuotteen tehtävälisan järjestämisen myös kehitystiimille.

**Kehitystiimi** koostuu asiantuntijoista, jotka kehittävät ja tuottavat jakeluun tuoteinkrementin, joka täyttää yhteiset *Definition of Done* laatuvaatimukset. Kehitystiimi ja vain ainoastaan kehitystiimi, päättää miten tuotteen tehtävälisasta luodaan inkrementtejä, joilla toimiva julkaisu saadaan aikaan. Vaikka tiimin jäsenillä on käytännössä aina erilaisia taitotasoja tai erilaisia osaamisalueita, tiimi on aina kokonaisuudessaan jaetussa vastuussa.

**Scrum Master** toimii palvelijana tiimille ja valmentaa Scrumin määritelmiä ja käytäntöä. Tiimin ulkopuolisessa vuorovaikutuksessa Scrum Master auttaa ymmärtämään toimintaa haittaavan ja edistävän vuorovaikutuksen eroja. Tuoteomistajalle Scrum Masterin roolissa keskeistä on tuotteen tehtävälisan läpikäynnin hallinnan tuki ja varmistaa, että tavoitteet, sisältö ja tuotealue on ymmärretty tarpeeksi kattavasti. Organisaation tasolla Scrum Masterin rooli on valmentaa Scrumin käyttöä.

Scrumiin kuuluu neljä säännöllisesti järjestettävää tapahtumaa. Tapahtumien tarkoitus on minimoida turhat Scrumiin kuulumattomat kokoukset pois ja tarjota kommunikaatioon säännölliset tavat. Tapahtuvat muodostavat mahdollisuuden tiimille tutkia ja muokata toimintaa tavoitteiden täyttämiseksi. Kaikki Scrumin tapahtumat ovat tiukasti aikarajoitettuja, joten niiden maksimiaika on ennalta määritelty.

Sprintti on enintään kuukauden pituinen aikakehikko, jonka aikana kehitystiimin on tarkoitus saada valmistettua yksi tuotteen tuotanto- ja julkaisukelpoinen inkrementti. Sprintit pysyvät vakiomittaisina kehityksen alettua ja uusi sprintti alkaa aina heti edellisen päättyttyä. Sprintin sisällä pidettävät tapahtumat ovat (engl. Sprint Planning) suunnittelukokous, päivittäiskokoukset (engl. Daily Scrums), katselmointitapahtuma (engl. Sprint review) sekä retrospektiivi (engl. Sprint Retrospective).

Sprint on rauhoitettu aika, jolloin sen sisältöön ei lähtökohtaisesti tehdä muutoksia eikä laatuvaatimuksia alenneta. Sprintillä on tavoite (engl. Sprint Goal), joka kuvaa mitä sprintin aikana yritetään saavuttaa. Tuoteomistajan kanssa yhteistyössä Sprintin sisältöä voidaan tarkentaa. Sprintin voi perua pelkästään tuoteomistaja yhdessä kehitystiimin kanssa. Peruutus tehdään jos liiketoiminnallisista syistä Sprintin tavoite muuttuu olennaisesti tai muista syistä sitä ei voida saavuttaa. Peruuttaminen ei saa olla toistuva käytäntö, koska se vaikuttaa radikaalisti tiimin arkeen, jossa pyritään keskittymään aina saavuttamaan Sprintin tavoite.

**Sprint Planning** suunnittelukokous on kuukauden sprintille maksimissaan kahdeksan tunnin aikarajoitettu tapahtuma. Tapahtumassa Sprintin sisältö suunnitellaan yhteistyössä koko tiimin kanssa. Onnistuneen Sprint Planning tapahtuman jälkeen ollaan yhteisymmärryksessä koko Scrum-tiimin kanssa siitä, mitä alkavassa Sprintissä valmistettava tuoteinkrementti sisältää. Sprintin tavoite ja päämäärä on tällöin lyöty lukkoon.

**Daily Scrum** on aikarajoitettu 15 minuutin päiväkokous kehitystiimille. Sen tarkoituksena on parantaa tiimin kommunikaatioita ja mitata edistystä kohti Sprintin tavoitetta. Tyypilliset kolme kysymystä joihin päiväkokouksessa pyritään vastaamaan: *Mitä tein eilen auttaakseni kehitystiimiä saavuttamaan Sprintin tavoitteen? Mitä teen tänään auttaakseni kehitystiimiä saavuttamaan Sprintin tavoitteen? Näenkö esteitä, jotka estävät minua tai kehitystiimiä saavuttamaan Sprintin tavoitteen?* Päiväkokous on varattu vain kehitystiimin keskinäiseen tiedonvaihtoon jolloin muut osallistujat eivät vaikuta tai osallistu keskusteluun.

**Sprint Review** eli katselmointitapahtuma pidetään Sprintin lopussa yhdessä sidosryhmien kanssa, joista tärkeimpänä tuoteomistaja. Tarkoituksena on läpikäydä kehitetty ja julkaistu tuoteinkrementti siltä osin mitä saatiin valmiiksi. Jos ilmeni ongelmia tai jotain ei saatu valmiiksi, käydään keskustelua mitä asioita kohdattiin, ja miten niitä voidaan välttää jatkossa. Tapahtuman lopputuloksena on päivitetty tuotteen tehtävälista ja käyty keskustelu tuoteominaisuuksista, jota voidaan käyttää suunnittelemaan myöhemmin seuraavaa sprinttiä.

**Sprintin retrospektiivi** on tapahtuma, jossa Scrum-tiimillä on mahdollisuus tutkia toimintaansa ja jonka pohjalta luoda suunnitelma parantamaan havaittuja epäkohtia seuraavien Sprinttien aikana. Scrum Master fasilitoi tapahtumaa ja varmistaa, että ilmapiiri on positiivisen rakentava. Retrospektiivin tarkoitus on tarkastella edellisen sprintin työtä suhteessa ihmisiin, ihmissuhteisiin, prosesseihin ja työkaluihin. Tunnistaa asioita jotka meni hyvin tai mitkä vaativat parannusta. Lopuksi luodaan suunnitelma parannuksiin, jolla Scrum-tiimi voi parantaa toimintaansa.

Scrum-viitekehyksen artefaktit esittävät olennaisen tiedon Scrum-tiimin suunnitellusta, käynnissä olevasta ja valmistuneesta työstä. Artefakteja voi olla enemmänkin ja tyypilli-

sesti niitä löydetäänkin enemmän prosessia parantaessa. Olennaista on, että kaikki osapuolet käyttävät ja ymmärtävät artefaktien sisällön. Minimissään kuitenkin seuraavat Scrumin artefaktit ovat käytössä:

Tuotteen tehtävälista (engl. **Product Backlog**) on järjestetty lista kaikista tuotteeseen tarvittavista asioista. Tuoteomistaja on vastuussa sen sisällöstä, saatavuudesta ja tärkeysjärjestyksestä. Lista sisältää kaikki ominaisuudet, toiminnot, vaatimukset ja parannukset sekä korjaukset, joita halutaan tehdä tulevissa versioissa. Tuotteen tehtävälistaa jalostetaan kehitystiimin ja tuoteomistajan toimesta koko kehityksen ajan (engl. **backlog refinement**). Jalostuksen tarkoituksena on syventää listan tehtävien arvoa ja työmääräarvioita, joka auttaa listan priorisointia sekä Sprint suunnittelua. Listan ylimmät, eli tärkeimmäksi priorisoidut tulevat yleensä valituksi seuraavalle Sprintille, joten yleensä ne ovat tarkemmin yksilöityjä ja arvioituja.

Sprintin tehtävälista (engl. **Sprint Backlog**) on joukko valittuja toteutettavia ominaisuuksia tuotteen tehtävälistalta, jotka sprintin suunnittelussa on arvioitu tuotettavaksi sprintin tuloksena. Näistä koostuu tuotteen julkaistava inkrementti sprintin päätteeksi. Vähintään yksi priorisoitava tehtävä retrospektiivistä valitaan sprintille, jotta toiminnan parantaminen varmistetaan. Sprintin tehtävälista tarjoaa reaaliaikaisen näkymän tiimin työhön ja sen edistymiseen.

Seurantänäkymä (engl. **Monitoring Sprint Progress, Burn down chart**) tuottaa reaaliaikaisen tiedon jäljellä olevasta arvioidusta työstä Sprintin kuluessa. Tällä voidaan seurata Sprintin tavoitteeseen pääsyn etenemistä ja kehitystiimi voi hallita työn etenemistä.

Sprintin inkrementti tarkoittaa kaikkien Sprintin aikana valmiiksi tehtävien yhteistä joukkoa, josta muodostuu julkaistava tuotteen osa. Tavoitteena on saada jokaisen sprintin valmistuttua inkrementti tuotteen toiminnollisuuksista julkaistua. Yksittäisen inkrementin valmistuessa on sen täytettävä valmiin määrittely (engl. **Definition of Done, DoD**). Määrittelyn tavoitteena on kehitystiimin ja muiden sidosryhmien kanssa ymmärtää mitä vaatimuksia valmiin tuoteinkrementin julkaisukelpoisuuteen kuuluu. Valmiin määrittelyn avulla kehitystiimi oppii työmääräarvioissa ottamaan huomioon kaiken mitä valmiiseen työhön pitää sisällyttää. Tuoteomistajalle määrittelyn täytyminen kertoo, että inkrementti sprintin tuloksena on julkaisukelpoinen. Scrum-tiimin toiminnan kehittyessä, määrittelyn tasoa voidaan syventää, koska ajan kuluessa aletaan tunnistamaan uusia laadullisia kriteerejä.

## 2.6.2 Kanban ohjelmistotuotannossa

Lean kirjallisuudessa alaluvussa 2.4 mainitun Toyota Production Systeminkin yksi menetelmä oli imuohjauksen käyttö valmistuksessa (engl. just-in-time manufacturing). Työkalu tai toimintatapa, jolla imuohjausta toteutettiin oli Kanban. *Kan-ban* semanttisesti viittaa japanilaiseen sanapariin, *kan* = *signaali* ja *ban* = *kortti*, eli *signaalikortti*. Valmistavassa teollisuudessa kanban-kortti lähetetään tuotannon edelliseen vaiheeseen kertomaan, että seuraavalla vaiheella on kapasiteettia jalostaa lisää tuotetta. Työntekijät edellisessä vaiheessa eivät lähetä puolivalmistetta oma-aloitteisesti eteenpäin, vaan valmistavat sitä puskuriin. Puolivalmisteen saa lähettää puskurista eteenpäin, kun on tiedossa signaalikortin vastaanoton myötä, että seuraava vaihe kykenee ottamaan tuotantoa lisää vastaan. Tätä prosessia hallitaan kanban-korteilla, jotka liikkuvat tuotannon linjastossa viestin välittäjänä. Signaalikorttien lisäksi menetelmän käyttöön kuuluu pohjimmiltaan myös Lean-ajattelun peruspilari *kaizen* eli toiminnan jatkuvaa parantaminen (Modig & Åhlström 2013; Torkkola 2015).

Kanbanin soveltaminen asiantuntijatyössä ja erityisesti ohjelmistotuotannossa on yleistynyt merkittävästi 2010-luvun taitteessa julkaistun kirjallisuuden myötä. Esimerkiksi David J. Anderson (2010) julkaisi kirjan, jossa kuvattiin miten valmistavan teollisuuden Kanbania sovelletaan asiantuntijalähtöiseen ohjelmistotuotantoon. Kanbanin käyttö asiantuntijatyöhön ohjelmistotuotannossa onkin lisääntynytkin merkittävästi yhä enemmän aina tähän päivään asti (Andersson 2010; VersionOne 2018). Viimeisimmät tutkimukset osoittavatkin, että on yhä edelleen nouseva trendi, että Scrum- ja XP-menetelmistä siirtyään Lean menetelmistä sovellettuun Kanbaniin (Wang et al. 2012; Al-Baik & Miller 2015). Corey Ladas (2008) *Scrumban* -kirja kuvasi miten Lean ajattelun Kanban istuu asiantuntijatyöhön ja erityisesti miten suositusta Scrum-tuotekehitysmenetelmästä voidaan siirtyä vaiheittain Kanbanin käyttöön. Tässä opinnäytetyössä Kanbania käsitellään jatkossa ohjelmistotyön kontekstissa. Seuraavat kappaleet tiivistävät Kanbanin käytön ja soveltamisen ohjelmistotuotantoon ja asiantuntijatyössä työtehtävien ja työtehtävöirran hallintaan.

Kirjassaan Andersson (2010) kertoo oivalluksen Kanbanin käyttämisestä asiantuntijatyöhön lähteneen siitä, että ohjelmistokehityksen elinkaaren eri vaiheita pystyi mallintamaan arvovirtana. Arvovirran mallintaminen oli jo Lean-ajattelusta tuttu menetelmä, jota sovellettiin valmistavassa teollisuudessa. Mallintamisen jälkeen virtausprosessi ja sen eri vaiheet voitaisiin visualisoida. Visualisoinnin avulla pystyi näkemään, miten kehitettävä sovelluksen osa eli inkrementti muodostui elinkaaren eri vaiheiden kautta aina julkaisuun asti. Työvirran visualisointi taas auttoi tunnistamaan systeemin pullonkauloja, jossa töiden virtaus hidastuu.

Kanban ei siis ole ohjelmistokehityksen elinkaaren metodologia tai projektinhallinnan työkalu. Kanban lähtee siitä, että on olemassa oleva prosessi, johon Kanbania sovelletaan ja jota lopulta lähdetään kehittämään Lean-ajattelun ja periaatteiden hengessä. Monien



muiden ketterien menetelmien vastakohtaisesti Kanbanissa ei siis sovelleta mitään tiettyä prosessin tarkkaa viitekehystä tai oteta tiettyjä metodeita käyttöön. Kanban lähtee siitä, että jokainen tuote, tiimi ja ympäristö on aina yksilöllinen ja omaa aina jo omannäköisensä prosessin. Tähän Kanban antaa työkaluja ja käytäntöjä, jolla tätä yksilöllistä prosessia lähdetään parantamaan, mutta minkään työkalun tai -tavan käyttö ei ole pakollista (Ladas 2008; Andersson 2010).

Kanbanista ei myöskään ole mitään yksittäistä absoluuttista kuvausta, käytäntöä tai ominaisuuslistaa, vaan vakiintunut alan kirjallisuus, tutkimukset ja alan vaikuttajat ovat luoneet eri tuotoksia, joiden pohjalta soveltaminen ohjelmistotuotantoon on yleistynyt. Tämä myös toisaalta hämmentää ja haittaa käyttöönottoa, koska käsitykset sisällöstä ja ominaisuuksista voivat vaihdella (Al-Baik & Miller 2015).

Kanbanin soveltamiseen ohjelmistotuotannossa lähteestä riippuen annetaan kolmesta viiteen olennaista lähestymistapaa. Seuraavaksi ne on esitetty kirjallisuudessa ja kirjallisuuskatsauksissa esitettyä järjestystä mukaillen (Kniberg 2011; Kniberg & Skarin 2010; Andersson 2010; Ladas 2008; Al-Baik & Miller 2015).

1. **Työvirran visualisointi** tarkoittaa työtehtävien vaiheiden esittämistä joko ohjelmallisesti tai perinteisellä fyysisellä taululla.

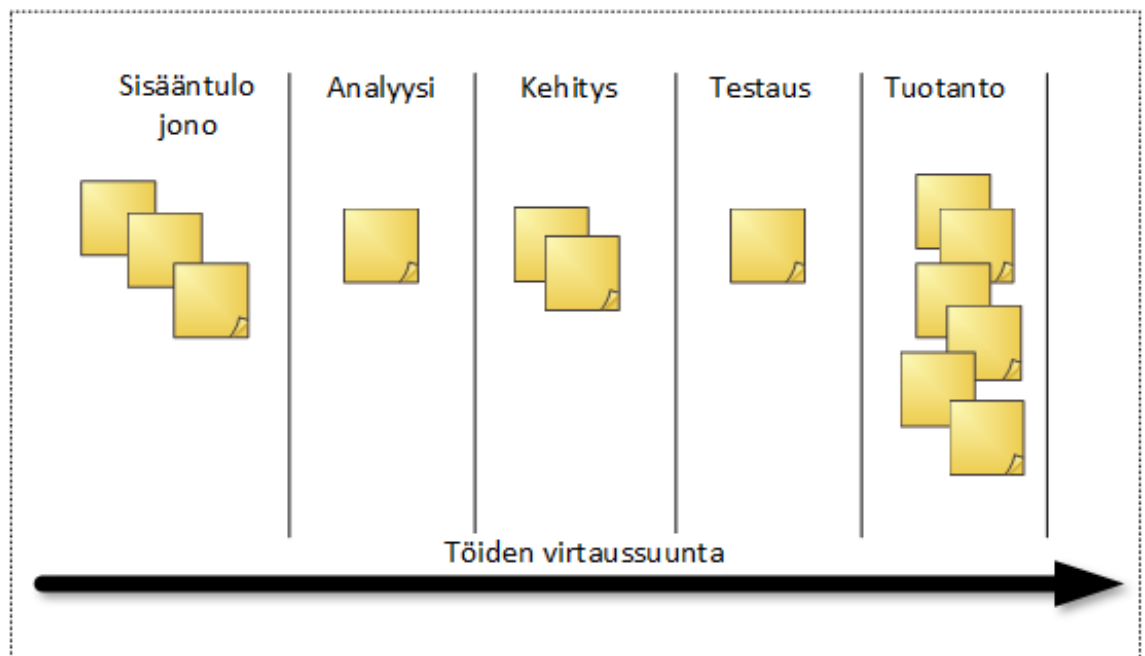
Kanbania käyttöönotettaessa tunnistetaan ensiksi tiimin arvovirtaketju. Arvovirtaketjun tunnistamisen avuksi on jo visualisoitu työvirtaus eri vaiheiden kautta. Organisaatioissa voi olla myös taipumusta painostaa avaamaan arvovirtaketju virallisten prosessien mukaan vaikka todellisuudessa tehtäisiinkin asioita näistä poikkeavalla tavalla. Tällöin on tärkeää kuvata prosessi juuri niiden epävirallisten, mutta totuuden mukaistenkin vaiheidenkin kautta. Olennaista on kuvata alussa nykytilaa, ei tavoitetilaa tai ideaalitilaa.

Andersson 2010 kirjan arvovirtaketjun kuvaus alkaa määrittelemällä prosessin alku- ja loppupiste. Pisteet ovat rajapintana töiden alavirran (engl. downstream) ja ylävirran (engl. upstream) sidosryhmiin. Esimerkkejä ylävirran sidosryhmästä voi olla tuoteomistajan vaatimukset tai tuotannon häiriöt. Alavirran sidosryhmät voivat liittyä toiminnallisuuden julkaisuun ja käyttöönottoon. Näiden löytämisen jälkeen määritellään työtehtävätyypit, esimerkiksi *käyttötapaus* (engl. *use case*), *ominaisuus*, *vaatimus*, *ohjelmointivirhe* (engl. *bug*) tai *tuotannon häiriö* (engl. *incident*). Erilaisilla työtehtävätyypeille on keskenään yleensä hieman eri työnkulku ja kiireysaste, joten tämä auttaa avaamaan näkyvyyttä prosessin sisään. Tämän jälkeen piirretään aktiviteettikaavio prosessista. Aktiviteettikaaviossa kuvataan aktiviteetit, joita työlle tehdään. Tämän perusteella voidaan tunnistaa, mitä työvaiheita käydään läpi ja muodostaa näistä sähköisiin tai fyysisiin perustuva Kanban-taulu.

Työtehtävät esitetään sähköisellä kortilla tai taululla muistilapulla, johon kuvataan lyhyesti työn sisältö. Korttia siirretään työvirran suunnaisesti sarakkeesta toiseen.

Pystypalkit kuvaavat työn elinkaaren vaiheita kuten esimerkkinä Kuva 3 esittää viisi eri vaihetta. Myöhemmin myös eri tehtävätyyppejä voi visualisoida omilla riveillään, jolloin ne muodostavat eri vaakatason ratansa (engl. swimlanes). Esimerkiksi tuotannon häiriöt pyritään korjaamaan nopeammin kuin kehityspyynnöt, joten ne voivat kulkea nopeammin omalla radallaan.

Kanbania käyttöönottaessa onkin tärkeä, että se otetaan käyttöön vaiheittain ja aluksi keskitytään visualisoimaan olemassa oleva prosessi. Prosessiin ei tehdä vielä muutoksia, vaan yritetään saada siihen ensiksi näkyvyys, jotta sen toimintaa voidaan ymmärtää ja myöhemmin parantaa. Tiimin totuttaminen työvirran visualisointiin ja tehtäväkorttien siirtelyyn päiväkokouksissa on jo arkisen toiminnan muutos itsessään, jolloin kaikkea muutosta ei pidä jalkauttaa kerralla, eikä vaikuttaa vielä itse asiantuntijan tekemiseen.



**Kuva 3 Esimerkki Kanban-taulusta, jossa työvirtauksen suunta vasemmalta oikealle. Pystysarakkeet nimetty työnkulun eri vaiheiden mukaan. Työtehtävistä tehdään kortti, jota päiväkokouksessa viedään työn edistymisen mukaan kohti oikeaa laitaa. Kuvassa voisi sisääntulojonossa olla esimerkiksi asiakkaan kehityspyyntöjä, jonka jälkeen ne analysoidaan ja aikanaan siirtyvät kehityksestä kohti tuotantoonvientiä.**

2. **Keskeneräisen työn rajoitusta** (engl. Work-in-Progress limit, WIP limit) aletaan soveltaa kun työprosessien visualisointi on tehty Kanban-taululle. Tarkoituksena on lähteä haarukoimaan rajaa työn enimmäismäärälle yksittäisessä (pystysarakkeen) vaiheessa.

Tällä pyritään tunnistamaan, ja myöhemmin estämään vaihtelu, joka seuraa liian monen rinnakkaisen asian tekemisestä, esimerkiksi estämällä vaiheeseen osallistuvien asiantuntijoiden ylikuormittumista. Jos kaikki prosessin resurssit ovat jo ylikuormittuneet, aletaan rajoittamaan lähtöpään virtausta. Rajoittamalla virtausta voidaan myös keinotekoisesti luoda lisää joutoaikaa, jota voidaan käyttää prosessin parantamiseen. Virtausta rajoittamalla on myös helpompi löytää pullonkauloja, kun nähdään mihin vaiheeseen työt kasautuvat. Tällöin kasautuneen vaiheen seuraava vaihe on systeemin sen hetkinen pullonkaula.

Keskeneräisen työn rajoitukselle ei ole olemassa mitään oikeaa lähtöarvoa tai laskentakaavaa. Ensimmäiseksi on lähdettävä kokeilemaan jotain, esimerkiksi tiimin henkilömäärästä pienempää arvoa. Arvoa muokataan myöhemmin arvon vaikutuksia analysoitaessa. Kanban ei siis määrittele miten rajoitusta käytetään, tai mitä sen ylittamisestä seuraisi. Raja-arvon ylitys voi jollakin tiimillä ja prosessilla olla tilanne, johon ei haluta päätyä, kun taas jonkun muun tiimin prosessissa se on heräte kokouksen koollekutsumiseen. Kokouksessa tiimi voi analysoida tilanteen, oliko ylitys perusteltu tai mitä on tapahtunut, jonka takia tilanteeseen päädyttiin. Tavoitetilana on päätyä arvoon, joka jokaisessa vaiheessa estää resurssien liiallisen ylikuormittumisen ja mahdollistaa tehokkaan virtauksen.

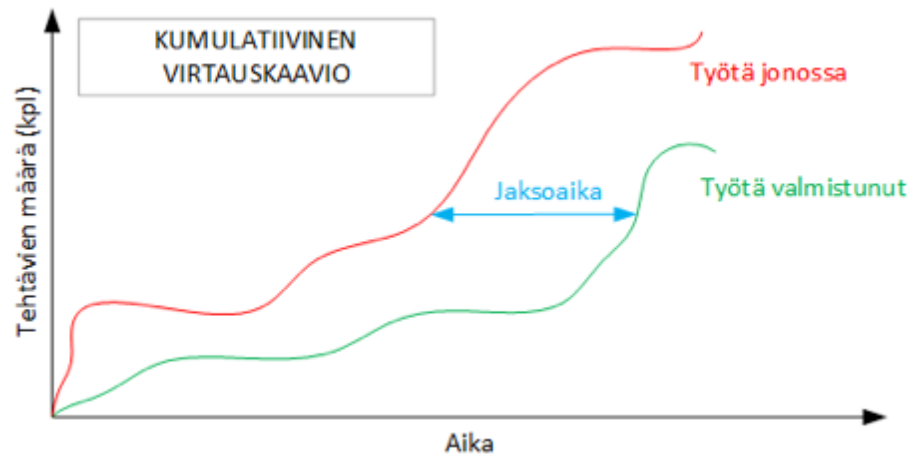
**3. Prosessin mittaus ja hallinta** ovat olennainen osa toiminnan kehittämistä. Tyyppillisiä mittareita työn virtauksesta ovat:

- a. Tehtävän jaksoaika (engl. cycle time), jota mitataan työn aloittamisesta sen lopettamiseen. Jaksoaika kertoo prosessista työn konkreettiseen tekemiseen käytettävää aikaa.

Tehtävän läpimenoaika (engl. lead time) on yleensä koko tehtävän tai tehtäväkokonaisuuden aika, esimerkiksi määrittelystä tuotantoon vientiin asti. Tehtävän läpimenoajan näkee ja kokee yleensä asiakas työntilajana. Kanban-taululla tämä tarkoittaisi tehtäväkorttien siirtymiseen kulunutta aikaa ensimmäisestä sarakkeesta viimeiseen. Tällöin mitataan Kanban-taululla visualisoidun prosessiin kulunutta aikaa ja oletus on, että tarpeen tunnistamisen jälkeen se on päätynyt heti tiimin työjonoon taululle.

Läpimeno tehokkuus (engl. through put) yleensä kertoo jonkin läpimenneen määrän suhteessa aikaikkunaan, esimerkiksi paljonko tehtäviä virtaa kuukauden aikana. Arvot voivat olla keskimääräisiä tai absoluuttisia. Läpimeno tehokkuutta kutsutaan myös velositeetiksi (engl. velocity).

Virtauksen jaksoaikaa voidaan mitata myös kumulatiivisesti, jolloin saadaan pitkällä aikavälillä näkymään jaksoajan kehitys kuten esitetään esimerkkinä Kuva 4.



*Kuva 4 Vaaka-akselilla esitetään aika ja pystyakselilla tehtävien määrä. Vaaka-akselilla on kaksi käyrää, kumulatiivisesti prosessin läpimenneet jonossa olleet (punainen käyrä) ja valmiiksi tehdyt työt (vihreä käyrä). Vertaamalla näiden vaaka-akselin erotusta, nähdään jaksoajan kehityksen ajan kuluessa (sininen käyrien aikojen erotus).*

- b. Jonopituudet (engl. queue lengths) ovat hyvä mittari vaihtelun havaitsemiselle ja visualisointiin. Jos jonopituuksissa on suurta vaihtelua, voi juurisyy olla kapasiteetin puute tai kysynnän vaihtelu. Syytä ei voida jonopituudesta suoraan johtaa, mutta jonopituus näyttää ongelman, jonka pohjalta voidaan aloittaa syiden etsintä.

Edellä mainittujen mittareiden nopean käyttöönoton sijaan onkin olennaista tunnistaa ja ymmärtää mitkä mittarit palvelevat ja kehittävät juuri sitä prosessia, johon Kanbania ollaan soveltamassa. Saadun ymmärryksen pohjalta voi lähteä mittaamaan asioita, joita myöhemmin muuttamalla, yritetään saada haluttu vaikutus prosessiin.

#### 4. Kehitä edelleen prosessia ja ymmärrä siihen vaikuttavat käsitteet

- a. **Kadenssi (engl. cadence)** on tahti, jolla työtehtävät etenevät prosessissa. Kanbanissa voidaan puhua jakelukadenssista (engl. delivery cadence), joka tarkoittaa sitä julkaisuväliä jolloin toimitamme valmiin tuotteen tai toiminnallisuuden ulos. Scrumin yhteydessä puhuttiin yhdestä sprintin mittaisesta inkrementistä, jossa tuotetaan yksi julkaistava tuotteen versio esimerkiksi 2-4 viikon välein. Kanbanissa jakelukadenssi voi olla mitä tahansa päivän ja äärettömän väliltä. Kadenssi voi olla säännöllinen tai

varta vasten vain pyynnöstä tehtävä julkaisu. Julkaisutahti määritellään lopulta osapuolten vaatimusten ja prosessin rajoitusten mukaan.

Sisääntulokadenssilla (engl. input cadence) tarkoitetaan tahtia, jolla tekemistä jonottavia tehtäviä priorisoidaan. Kanban ei määrittele Scrumin tyyppisesti inkrementtiä eikä erillisiä suunnittelupalavereita, vaan prosessikohtaisesti etsitään sopiva heräte priorisoimaan sisääntulevat työt.

- b. **Pullonkaulateoria (engl. theory of constraints).** Leanin yhteydessä mainitun pullonkaulateorian mukaan systeemissä on aina pullonkaula jossain. Poistamalla pullonkaula, tulee esiin seuraava virtausta hidastava pullonkaula. On hyvä ymmärtää, että kaikkia pullonkauloja ei voi ikinä poistaa, mutta sen sijaan keskittyä tunnistamaan ne, jotka haittaavat virtauksessa sen hetkiseen tavoitetilään pääsyä.
- c. **Rooleja** ei Kanbanissa ole määritelty keskeisimmässä kirjallisuudessa, koska tarkoitus on parantaa jo olemassa olevaa prosessia. Käytännössä olemassa olevia rooleja on prosessin kautta, esimerkiksi työtehtävien suorittaminen vaatii asiantuntijatasen rooleja sekä tehtävistä riippuen erilaisia osaamisia. Työtehtäviäluokkien ja työtehtävien priorisointi tapahtuu niiden asiakasomistajuuden mukaan, joten liiketoiminnan edustajan roolistakin prosessiin tulee riippuvuuksia. Kanban ei kuitenkaan käsittele niitä, vaan työprosessi ja sen kehitys ovat keskiössä.
- d. **Rituaaleita** tai pakollisia tapaamisia ei ole määritelty suoraan Kanbanissa. Käytännössä kuitenkin työnvirtauksen visualisointi vaatii sen jatkuvaa päivittämistä ja tilanteen yhteistä läpikäyntiä jossain muodossa. Tyypillisin mainittu tapa on päiväkokoukset (engl. daily stand up), jossa nopeasti esimerkiksi työpäivän alussa kokoonnutaan Kanban-taulun ääreen ja käydään läpi tilanne tiimin töihin. Jatkuvaan toiminnan parantamiseen ja kehittämiseen on käytetty retrospektiivikokouksia. Suunnittelu, priorisointi ja julkaisu vaativat käytännössä myös tiimin yhteistoimintaa, mutta Kanban ei määrittele miten ne tulee toteuttaa.
- e. **Palvelutasoluokat (engl. service level class)** määrittelevät eri tyyppiseen palveluun liittyvän palvelutason. Esimerkiksi laajavaikutteinen ja kriittinen tuotantohäiriö määritellään paljon kiireellisempään luokkaan kuin uuden tuotetta laajentavan ominaisuuden käyttöönotto. Palvelutasoluokalla voidaan määritellä priorisointisäännöt miten tehtävä etenee. Palvelutasoluokka on myös hyvä visualisoida tehtäväkorttiin, jotta tärkeimmät tehtävät erottuvat koko työtehtäväjoukosta. Visualisoinnin lisäksi Kanban-taulun vaakarivit, eli ns. uimaradat (engl. swimlines) voidaan haluttaessa jaotella eri palvelutasoluokkien mukaan. Tällöin kiireellisemmin tehtäväkortit etenevät erillään kuin vastaavat kiireettömät eri rivillä. Tällä varmistetaan ja visualisoidaan, että koko virtauksen kannalta kiireellisemmän

työt virtaavat nopeammin. Palvelutasoluokan määrittäminen prosessiin voi vapauttaa myös johdon aikaa kehittämään prosessia, sen sijaan, että katsotaan kiireellisen työn perään. Palveluluokalla olisi hyvä olla tavoitteellinen läpimenoaika (engl. target leadtime), jotta toimitusaikatäsmällisyys noudattaa määriteltyä tavoitetta ja toimitusaikapyyntöjä pystytään sopimaan.

Palvelutasoluokitus auttaa myös Ladasin (2008) havaintoon, jossa usein kehitystiimeillä on tilanne, että erilaisia työtehtävätyyppejä pitäisi jakaa eri tiimien resurssien välillä. Yksi tapa ratkaista asia olisi jakaa henkilöt täysin tiettyjen osaamisalueiden mukaan. Toinen vaihtoehtoinen tapa on yrittää ratkaista asia niin, että kaikki osaavat kaikkea. Kumpikaan ei yleensä ole mahdollista, mutta tähän Kanban mahdollistaa hybridiratkaisun, jossa kapasiteettia voidaan rajoittaa vain tietyille tehtävätyypille, ilman osaavien henkilöiden suoraa sitomista tehtävätyyppeihin.

#### *Muutos Scrum-menetelmästä kohti Kanbania:*

Tiimillä voi olla jo kokemusta Scrum-menetelmästä, mutta Kanbania ja sen käyttöä ohjelmistotuotannossa ei välttämättä tunneta. Puhekielessä Scrumban-menetelmä voi olla edellisessä aluvuossa esitellyn Scrumin ja Kanbanin eri ominaisuuksien yhdistelmä tai vielä useamman ketterän menetelmän yhdistelmä. Scrumban nimenä on tietävästi saanut alkunsa Corey Ladasin kirjasta, joka ensimmäisten joukossa esitteli Kanbanin käyttöönottoa ohjelmistotuotannossa (Ladas 2008). Corey Ladas esittää vahvan näkemyksen siitä, että Scrum pahimmillaan piilottaa esimerkiksi tuoteomistajan taakse todellisia sovelluksen käyttötarpeita, kun taas Lean-ajattelu kehityksessä korostaa päästä päähän näkyvyyttä ja arvoketjun luontia. Ladas viittaa myös siihen, että kehittäjäyhteisö on lukkiutunut liian kapeasti Scrumin kuvaamien rajojen sisällä toimimiseen. Kirjassaan hän lähtee kuvaamaan vaihe vaiheelta, miten Scrum-menetelmää käyttävä tiimi voi pienin askelein ottaa yhä enemmän Kanbania käyttöön, ja lopulta päätyy täysin imuohjattuun menetelmään virtaustehokkuuden maksimoiseksi. Seuraavaksi esitetään Ladasin esittämät vaiheet tiivistetysti eli miten Scrumia käyttävä tiimi voi muokata prosessinsa asteittain kohti Kanbanin imuohjausta:

1. Yksinkertaisuuden ja vähäisten muutosten vuoksi aloitetaan pitämällä Scrum sprintit ja niiden suunnittelupalaverit ennallaan. Kanbania aletaan rakentaa pala palalta Scrumin kehitysprosessin ja sprintin sisään.
2. Asetetaan aluksi tiimin jäsenkohtaisen rinnakkaisen työn rajoitus. Rajoitus voi olla esimerkiksi näin: *Jos ensimmäistä tehtävää ei voi edistää, voit aloittaa toista tehtävää, mutta et yhtään enempää.* Tällöin saataisiin kolmen hengen tiimille rajaksi kuusi rinnakkaista työtä (WIP). Tällä on aluksi tarkoitus totuttaa tiimille keskeneräisen työn rajoitusta uutena käsitteenä siirryttäessä kohti Kanbania.
3. Tehtävän kiinnittäminen henkilölle kannattaa Kanbanissa tehdä mahdollisimman myöhäisessä vaiheessa. Jos tehtävät kiinnitetään henkilöille jo Scrum sprinttiä

suunnitellessa, ovat ne jo valmiiksi jonossa tietyn henkilön tekemiseen, eivätkä edesauta imuohjauksen syntymistä tai tasaisen työkuorman leviämistä. Tehtävien odottaessa viimeistä mahdollista hetkeä tekijän kiinnitykselle, toteutuu töiden tasainen jakautuminen, joka taas vähentää odottelua. Myöhäinen kiinnittäminen tulee myös osaamiskapeikkojen vähentämistä, varsinkin jos tiimillä on taipumus kiinnittää tehtävät aina tehtävän aluetta hyvin osaavalle.

4. Lisätään jonoa varten ylimääräinen sarake tiimin tehtävätaululle. Esimerkiksi työtä odottavien (engl. TODO) ja työn alla olevien (engl. In Progress) sarakkeiden väliin. Tähän jonosarakkeeseen siirretään ne tehtävät, joilla on tehtävälistalla korkea prioriteetti. Edelleenkin näitä tehtäviä ei vielä kiinnitetä kellekään henkilölle. Lisäksi jonosarakkeeseen luodaan pieni keskeneräisen työn raja, jotta siinä pysyy vain aidosti tärkeimmät tehtävät. Tässä kohtaa olemme luoneet jo yksinkertaisen Kanban-järjestelmän. Meillä on edelleen käytössä Scrum sprintit ja suunnittelupalaverit, mutta Ladas (2008) kutsuu jo tämän vaiheen toteutusta *Scrubaniksi*. Nyt tiimille on muodostunut kapasiteetin ja imuohjauksen ensituntuma, koska he seuraavat keskeneräisen työn rajoitinta ja ottavat töitä jonosarakkeesta.
5. Tiimin käyttämä Scrumin seurantakäyrä (burndown) kertoo toimitetun arvon määrästä suhteessa Scrum sprinttiin, mutta ei juurikaan sitä, miksi esimerkiksi prosessi hidastelee. Tämän takia pyritään jatkossa kirjaamaan tietoa tehtävien määrästä, aloitus- ja lopetusajoista, jotta voidaan muodostaa kohdassa Kuva 4 esitelly kumulatiivinen virtauskaavio. Prosessin kehittyessä tunnistetaan yhä enemmän pullonkauloja ja kiinnitetään huomiota läpimenoaikoihin. Lopulta aletaan ymmärtää, että seurantakäyrän kertoma on vain seurausta jostain oireesta, esimerkiksi pullonkaulojen tuomasta hitaudesta, mutta läpimeno- ja jaksoajat ovat oireen syy. Näiden pohjalta on helpompi etsiä vielä tarkempaa juurisyytä prosessin kehittämiseksi.
6. Seuraavaksi Sprintin tehtävälistalle asetetaan kiinteä tehtävämäärän rajoitus, joka perustuu tiimin tehtävien läpimenoaikaan. Esimerkiksi jos tiimin Scrum sprintin pituus on kymmenen päivää ja yksi tehtävä ottaa päivän, tällöin tehtävälistan koko on syytä olla vähintään kymmenen. Nyt jos tehtävälistan koko oli kymmenen, asetetaan suunnittelukokouksen aloittamisen herätteeksi raja viiteen tehtävään. Tällöin tekemättömien töiden laskiessa alle viiden, aloitetaan suunnitteluprosessi, jolla saadaan tiimin tehtävälistalle lisää tehtäviä. Tämä mahdollistaa luopumisen Sprintin suunnittelukokouksesta ja tehtävälistan läpikäynnistä tulee herätepohjainen sen sijaan, että se olisi kalenterissa sidottu aina sprintin alkuun.
7. Varmuuden lisääntyessä on lopulta aika siirtyä kokonaan imuohjaukseen. Scrumin aikajaksotukseen pohjautuvat tapahtumat voidaan lopettaa ja sovittaa kehitys todellisten tarpeiden mukaan. Nyt ollaan saavutettu ideaali imuohjattu tehtävien virtausjärjestelmä, jossa suunnittelua ja julkaisua tehdään tapahtumalähtöisesti,

eli aidon tarpeen vaatiessa eikä menetelmän aikaikkunan ohjaamana. Ideaalitilanteen suunnitteluprosessin pitäisi tuottaa tiimille vain seuraavat tärkeimmät tehtävät, ei vähempää eikä enempää.

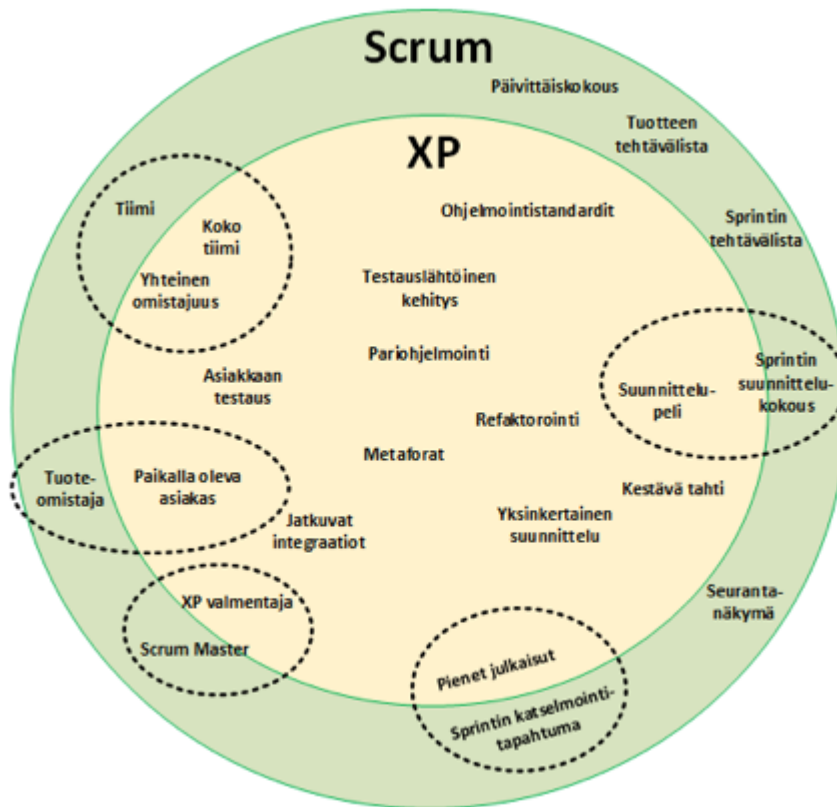
### 2.6.3 eXtreme Programming ohjelmistokehityksessä

Ketterän kehityksen viitekehyksiä ja menetelmiä on kehitetty lukuisia muitakin, joista edellä esiteltyjen jälkeen yksi käytetty on eXtreme Programming (XP). Kyselytutkimuksen mukaan vuonna 2017 pelkkää XP-menetelmää käytti vain 1% vastaajista, silti yksittäisiä XP-menetelmän käytäntöjä on otettu vieläkin käyttöön. Kuitenkin Scrum- ja XP-menetelmän hybridiä ilmoitti käyttävänsä jo 6% vastanneista (VersionOne 2018).

Extreme Programming on ohjelmistokehitykseen Kent Becking 90-luvulla esittelemä menetelmä (Beck & Andres 2006). Ketterien menetelmien tavoin XP pohjaa myös arvoajatteluun, jotka ovat *yksinkertaisuus*, *kommunikaatio*, *palaute*, *rohkeus* ja *kunnioitus*. Extreme Programming on kehittynyt ja levinnyt rinnakkain Scrum-menetelmän kanssa ja käytännössä sisältävätkin paljon samoja ominaisuuksia.

Esimerkiksi XP-menetelmän paikallinen tai paikalla oleva asiakas (engl. on-site customer) on hyvin lähellä Scrumin tuoteomistajan (PO) roolia. Kumpikin korostaa pohjimmiltaan roolia, joka tuo kehitykseen mahdollisimman läheisesti liiketoiminnalle arvoa tuottavat ja oikeat tarpeet. Kuva 5 esittää mukaillen Henrik Knibergin näkemyksen, jossa Scrum-menetelmä on kerroksena eXtreme Programming menetelmän ympärillä ja osa ominaisuuksista voidaan nähdä tarkoitukseltaan samanlaisena.





**Kuva 5** Henrik Knibergin näkemys Scrum- ja XP-menetelmien keskinäisestä suhteesta ja niiden yhtenäisistä ominaisuuksista (mukaillen Kniberg 2011, chapter 17.4).

Extreme Programming edellisistä esitellyistä menetelmistä poiketen määrittelee lisäksi ohjelmistokehitysprosessiin liittyviä tiimin sisäisiä käytäntöjä (Kniberg 2011):

- *Jatkuvassa integraatiossa* (engl. Continuous Integration) luodaan ja hallitaan järjestelmää, joka automaattisesti kääntää, integroi ja testaa kehitettävää tuotetta.
- *Pariohjelmoinnissa* (engl. Pair Programming) ohjelmointia tehdään parityöskentelynä yhdellä työpisteellä, jolla maksimoidaan oppiminen, parannetaan laatua ja minimoidaan virheet.
- *Testilähtöisessä kehityksessä* (engl. Test driven development) kehitetään ensiksi ohjelmakoodi, joka testaa luotavan tuotteen ohjelmakoodin osan. Tämän jälkeen kehitetään itse testin läpäisevä varsinaisen sovelluksen osa.
- *Yhteisen ohjelmakoodin omistuksessa* (engl. Collective code ownership) tiimiä rohkaistaan käsittelemään sovelluksen lähdekoodia yhteisenä. Jokaisella yksilöllä on mahdollisuus muokata mitä tahansa osaa sovelluksesta. Tällä pyritään luomaan helppoa ja yhdenmukaista suunnittelua koko sovellukselle.
- *Inkrementaalisen suunnittelun käyttämisessä* (engl. Incremental design improvement) ison ja valmiin suunnitelman sijaan, tehdään alussa pieni ja toimiva sovellus, jota uudelleensuunnittelulla laajennetaan iteratiivisesti.

## 2.6.4 Muut menetelmät ja menetelmien skaalaus

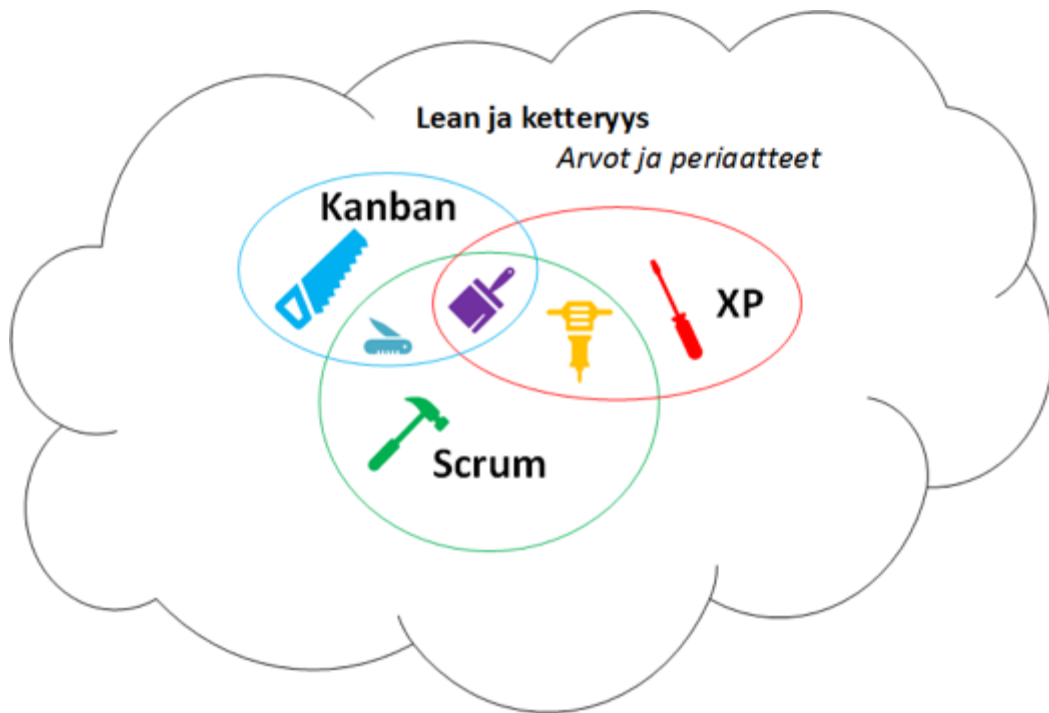
Edellä esitettyjen käytetyimpien menetelmien lisäksi alalla on myös muita vähemmän tunnettuja menetelmiä, mutta osaltaan ketterän kehityksen periaatteita sivuavia kuten Rational Unified Process (RUP), Dynamic System Development Method (DSDM), Crystal Methods ja Feature-Driven Development (Leffingwell 2008). Vuoden 2017 tehdyn ketterien menetelmien käyttötutkimuksen perusteella nämä eivät enää ole itsenäisinä menetelminä kovinkaan käytettyjä, mutta niistä voi olla poimittu jotain yksittäisiä ketteriä toimintatapoja vastaajien käyttöön (VersionOne 2018).

Suurissa tuotekehitysprojekteissa, jossa kaksi tai useampaa tiimiä kehittää samaa sovelusta tai keskenään tiiviisti integroituvia sovellustoiminnallisuuksia, on tullut tarve hallita kokonaisuutta laajemmin. Siinä missä esimerkiksi Scrum-viitekehys keskittyy yhden tiimin tuotekehitykseen, on menetelmiä ja viitekehyksiä, jolla hallitaan laajempaa useamman tiimin toimintaa tuotekehitysprosessissa. Näitä menetelmiä kutsutaan skaalaaviksi kehityksen menetelmiksi. Näiden skaalaavien menetelmien käyttöastetta tutkivan kyselytutkimuksen perusteella jopa 29% vastaajista olivat ottaneet käyttöön Scaled Agile Framework (SAFE) -viitekehityksen (VersionOne 2018). Scaled Agile Framework on laaja moniin hyväksi havaittuihin käytäntöihin ja periaatteisiin pohjautuva viitekehys, joka ei tarkastele enää pelkästään tiimitason toimintaa, vaan laajimmillaan toteutettuna usean tiimin, tuotteen, tuotehallinnan, budjetoinnin ja portfolion käsittelyä (Leffingwell 2019).

## 2.7 Yhteenveto tiimityöstä kohti käytännön menetelmiä

Tässä luvussa esiteltiin ohjelmistotuotannon asiantuntijatyö, tiimityön ja sen johtamisen näkökulmia sekä perusteltiin miksi ja miten ketterät menetelmät vastaavat tämän hetken vaatimuksiin tuottaa ohjelmistoja. Näiden jälkeen esiteltiin yksityiskohtaisemmin ohjelmistokehityksen yksittäisiä menetelmiä.

Alaluvussa 2.4 esiteltiin Lean-ajattelu ja alaluvussa 2.5 ketterän ohjelmistokehityksen arvot ja periaatteet. Nämä yhteistyössä yhdistettynä asiantuntijoiden tiimityöhön luovat pohjan käytettäville menetelmille ja toiminnan kehittämiseksi. Keskenään ketteryys ja Lean -ajattelu periaatteineen ovat kaksi toisiaan täydentävää joukkoa. Yksittäiset menetelmät, kuten esimerkiksi Scrum, Kanban ja eXtreme Programming voidaan nähdä konkreettisena tapana miten näitä arvoja ja periaatteita sovelletaan käytäntöön. Tämä havainnollistuu kohdassa Kuva 6, jossa esitetään Kniberg (2011) mukailleen esitetty menetelmät työkaluina toteuttamaan Leanin ja ketteryyden periaatteita käytäntöön.



**Kuva 6** Henrik Knibergin kirjaa *Lean from the Trenches* mukaillen (Kniberg 2011, chapter 17). Lean ja Agile ovat kaksi yhteensopivaa ja toisiaan täydentävää arvojen ja periaatteiden joukkoa, jotka luovat perustan menetelmille. Scrum, eXtreme Programming ja Kanban ovat konkreettisia työkaluja, joilla sovelletaan nämä periaatteet käytäntöön.

Toisaalta Wang et al. (2012) kirjallisuuskatsauksessa tutki Lean menetelmien käyttöä ketterässä kehityksessä, jossa Leanin ja ketteryyden määritelmien yhteydessä todettiin, että monesti kirjallisuudessa nämä käsitteet menevät sisällöltään ristiin sekä ovat lyhyesti määriteltyjä ja moniulotteisia. Huolimatta käsitteiden sisällön tarkasta alkuperän luokitelusta, voidaan silti todeta, että keskiössä ovat tiimi ja ihmiset. Nämä korostuvat jokaisessa menetelmässä. Tässä luvussa tiimityön näkökulmia avattiin ensimmäisissä alaluvuissa organisaatiopsykologian näkökulmasta, joista keskeisinä tunnistettiin hyvin suoriutuvan tiimin luominen, sen motivoimisen sekä johtamisen yhteisten jaettujen tiimitavoitteiden ja itseohjautuvuuden luomisen myötä. Nämä täysin samat näkökulmat tulevat esiin ja jopa korostuvat myös ketteryyden periaatteiden ja Lean kirjallisuuden kautta.

Nykypäivän ohjelmistoteollisuudessa on huomattu, että teollisen ajan johtamisen tarkat kontrollit ja rakenteet eivät ole luontevia osaavien asiantuntijoiden ja merkittävän liiketoiminta-arvon tuottamiseen. Kuten Ladas (2008) tähän esittääkin, paras tapa hallita riskiä ja epävarmuutta monimutkaisissa kokonaisuuksissa, on vastuun jakaminen sen keskittämisen sijaan. Ketterät periaatteet korostavat juuri tätä vastuun jakamista ja keskitetyn hallinnan vähentämistä. Tiimit eivät kuitenkaan voi aina päättää kaikkia asioita itse, ja toisaalta myös keskitettyä päätöksentekoaakin tarvitaan. Ladas (2008, chapter; Between Kanban and Pair ...) kuvaileekin keskitetyn ja jaetun johtamisen jaon ongelmaa näin:

*Miten hallita kommunikaatiota korkean osaamistason asiantuntijoiden välillä, jotka työskentelevät yhteisen ongelman äärellä, jota kuitenkin kukaan yksilönä ei voisi täysin ymmärtää?*

Johtajuuden jakaminen tiimeille tai osittain keskittäminen on aina riippuvainen kontekstista. Arvo tai periaate ei voi antaa asiaan täyttä vastausta. Samoin menetelmien ja periaatteiden käytön ristiriitajakin voidaan halutessa nähdä. Esimerkiksi eXtreme Programming -menetelmän pariohjelmointi kokemuseräisesti tuottaa parempaa laatua, mutta vaatii kahden asiantuntijan yhtäaikaisen panoksen. Tällöin tuotannon kannattavuuden kannalta, kaksinkertainen resurssikustannus tulisi myös lisätä riittävästi laatua. Jos tämä taas ei toteudu, Lean ajattelua vahvasti tulkiten, voisi kahden asiantuntijan käyttäminen olla tällöin osaltaan hukkaa, jota taas pitäisi pystyä nimenomaan poistamaan kehitysprosessista Lean-ajattelulla. Tällöin päätös pariohjelmoinnin käytöstä pitää aina tehdä tarkasteltavan kontekstin perusteella ja kokonaisuus ymmärtäen.

Menetelmät korostavat niiden mukauttamista käytäntöön, koska jokaisen tiimin ja kontekstin tapa käyttöönottaa asioita on yksilöllinen. Jokaisen organisaation täytyykin osata katsoa omia prosessejaan, projektin tai palvelun tavoitteita ja käytännön rajoitteitaan, jotta se osaa soveltaa Lean ja ketteryuden periaatteita ja työkaluja käytäntöön (Wang et al. 2012). On tärkeää ymmärtää, että seuraamalla tiettyä menetelmää tarkasti, ei voida varmasti saavuttaa ketteryyttä tai poistaa järjestelmällisesti hukkaa. Ymmärtämällä arvot ja periaatteet, joiden pohjalta menetelmät tarjoavat ratkaisua, tällöin vasta syntyy ymmärrys soveltaa työkaluja ja menetelmiä ongelmanratkaisuun.

### 3. TUTKIMUSSTRATEGIA JA -MENETELMÄT

Tässä luvussa kuvataan tutkimuksen strategia ja mitä menetelmiä tutkimuksessa on käytetty. Alaluvuissa esitellään valittujen menetelmien perustelut sekä soveltaminen lyhyesti. Lopuksi on esitelty tutkimushaastattelun teemojen valinta ja tulosten analysointi.

#### 3.1 Tutkimusstrategian käytäntöön pano

Tutkimusstrategialla tarkoitetaan kokonaisuutta, joka koostuu valituista tutkimusmenetelmistä tutkimuskysymyksen selvittämiseen. Hirsjärvi et al. (2007, s. 130) mukaan kolme perinteisen tutkimusstrategian jaottelua ovat sisältöineen seuraavat:

1. *Kokeellisessa tutkimuksessa* mitataan tietyn muuttujan vaikutusta toiseen muuttujaan sekä vaihdellaan mitattavan muuttujan olosuhteita lisätiedon saamiseksi.
2. *Survey-tutkimuksessa* kerätään valitulta joukolta tietoa haastattelu- tai kyselytutkimuksilla vakioмуotoisesti, jonka perusteella pyritään kuvailemaan ilmiöitä. Tämä sopii esimerkiksi laajalle valitulle kohderyhmälle tehtävään tiedonkeruuseen.
3. *Tapaustutkimuksessa (engl. case study)* tutkitaan yksityiskohtaisesti tiettyä tapausta tai pienen joukon tapauksia toisiinsa.

Tämän työn ensisijaiseksi tutkimusstrategian menetelmäksi on siten valittu *tapaustutkimus*, koska tarkasteltava ongelma täytti tapaustutkimuksen tyypilliset piirteet. Esimerkiksi Hirsjärvi et al. (2007) mukaan tyypillisiä tapaustutkimuksen piirteitä ovat yksittäinen joukko, esimerkiksi ryhmä tai tässä tapauksessa tiimi, jossa kiinnostuksen kohteena on esimerkiksi prosessit ja miten tätä tutkittava joukko on yhteydessä ympäristöönsä. Näiltä tutkimuksen kohdejoukoilta, esimerkiksi asiantuntijatiimin jäseniltä, kerätään haastatteluin ja havainnoin tietoa. Tutkittavan joukon ympäristönä voidaan pitää tiimin ympäröivää kontekstia kuten tässä organisaatioita, tehtäväkenttää ja asiakasta.

Hirsjärvi et al. (2007, s. 134) viitaten (Yin 1984) mukaan kartoittavan tutkimuksen tarkoituksena on *katsoa mitä tapahtuu, etsiä uusia näkökulmia ja löytää uusia ilmiöitä, selvittää vähän tunnettuja ilmiöitä ja kehittää hypoteeseja*. Täten tutkimusstrategian valintaa ohjasi tutkimusongelman ratkaisu kartoittamalla. Tutkimus luokitellaan kartoittavaksi, jos se pyrkii ilmentämään vaihtelevia ilmentymiä. Tiimin työnohjausmenetelmien valinta voidaan nähdä ilmiönä, jossa vaihtelevuutta tuo aina jokaisen tuotettavan palvelun, asiakkuuden sekä tiimin yksilöllinen konteksti.

Lisäksi tämän työn strategiaan kuului kirjallisuuteen perehtyminen, joka on oleellinen vaihe ennen kokeellista tutkimusta (Hirsjärvi et al. 2007; Hirsjärvi & Hurme 2008). Tutkimuksen teoreettinen viitekehys on rakennettu tunnistamalla tiimityöhön liittyviä sosiaalisia- ja psykologisia tekijöitä alan tutkimuksista sekä etsimällä käytetyimmistä tietotekniikka-alan työn- ja prosessinohjaus menetelmistä keskeisimmät. Lisäksi valintaan vaikutti kirjoittajan oma pitkä kokemus toimialalta ja tiedot tutkittavan palvelun tiimien käyttämistä malleista, joiden pohjalta koostettiin teoriaa haastatteluja silmällä pitäen.

### 3.2 Tiedonkeruu teemahaastattelulla

Ensisijainen valittu tapaustutkimuksen tiedonkeruutapa oli palvelun asiantuntija- ja johtajaroolissa toimivien haastattelut. Haastattelun käyttöä puoltavia etuja tiedonhankintamenetelmänä, jotka kirjallisuudessa mainitaan ja tunnistettiin tähän työhön sopivaksi (Hirsjärvi & Hurme 2008):

- Tutkitaan *vähän tai lähes tuntematonta aluetta*
  - Kohdepalvelussa ja sen yksilöllisessä kontekstissa on otettu käyttöön tiimimalli asteittain tutkimusvuoden 2018 aikana.
- Halutaan sijoittaa haastateltavan puhe *laajempaan kontekstiin*
  - Halutaan tunnistaa yhteisiä vaikuttimia tiimin työnohjaukseen koko tuotettavan palvelun laajuudesta
- Halutaan tarvittaessa *syventää* tietoja
  - Valittujen menetelmien lisäksi, halutaan ymmärtää ja etsiä syitä työnohjauksen menetelmien käyttöön ja tarpeisiin
- Haastattelua voi käyttää myös kartoitukseen ja sillä voidaan löytää uusia merkityksiä ja luoda hypoteeseja

Myös haastattelun haittoja pohditaan tämän raportin tutkimustulosten pätevyyttä ja luotettavuutta käsittelevässä alaluvussa.

Haastattelut toteutettiin teemoitettuna ja kohdennettuna yksilöhaastatteluna. Teemahaastattelua kutsutaan kirjallisuudessa myös puolistrukturoiduksi haastattelumenetelmäksi. Teemahaastattelu termi ei myöskään sido haastattelua kvalitatiiviseen tai kvantitatiiviseen, vaan yksinkertaisesti kertoo että haastattelu etenee tiettyjen teemojen mukaan (Hirsjärvi & Hurme 2008).

Haastatteluun suunniteltiin ennalta määritellyt teemat, joiden pohjalta tietoa kerättiin. Teemojen sisällä kysymyksillä ei ole tiettyä tarkkaa muotoa ja järjestystä, niiden tarkoitus on pääasiallisesti herättää haastateltava kertomaan subjektiivisia näkemyksiään teeman ympäriltä.

Aineiston ja tiedonkeruun määrä rajattiin valitun kohdepalvelun asiantuntijatiimeihin ja niiden johtamiseen. Rajausta tehtiin tutkittavan yrityksen kohdepalveluun, koska haluttiin

selvittää vaikuttavia seikkoja juuri tässä kyseisen toimittamisen kontekstissa ja hyödyntää tutkimustuloksia tämän asiakkuuden ja palvelun kehittämiseen. Aineiston määrä voidaan katsoa riittäväksi, kun vastaukset alkavat satureitua, eli niissä alkavat asiat toistuvat. Tämän toistuvuuden varmistamista tukee se, että jokaiselta haastateltavalta tiimiltä kysytään saman teeman kysymyksiä (Eskola & Suoranta 1998).

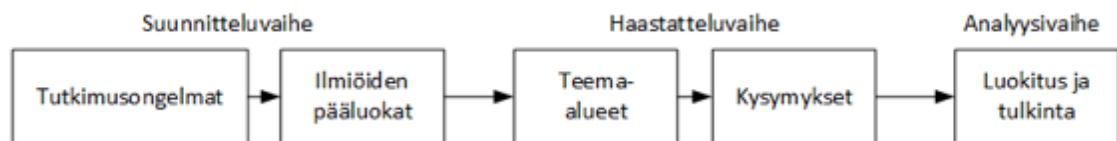
### 3.2.1 Haastatteluaineiston käsittely ja analyysi

Haastattelut on tallennettu, jonka jälkeen muistiinpanot on täydennetty tarkemmalla litteroinnilla. Litteroinnin yhteydessä vastaukset on järjestetty tarkempien teema-alueiden mukaan.

### 3.2.2 Teema-alueiden valinta

Tutkimushaastattelujen teema-alueiden paikka tutkimuskokonaisuudessa sijoittuu haastattelu- ja analyysivaiheeseen Kuva 7 mukaisesti (Hirsjärvi & Hurme 2008). Tämän työn tutkimusongelman valinnan jälkeen tunnistettiin ongelmaan liittyvät ilmiöiden pääluokat, etsittiin niistä keskeistä kirjallisuutta ja tutkimuksia. Näiden läpikäynnin perusteella voitiin tunnistaa ongelmaan liittyvät pääteemat. Suunnittelun apuna käytetty miellekartta on tämän työn liitteenä.

Pääluokat eli valitut pääteema-alueet toimivat apuna haastattelussa luettelomaisina asioina, joilla haastattelija voi ohjata haastattelua ja ohjata keskustelun pysymään aiheessa. Haastattelutilannetta varten oli myös johdettu tarkempia kysymyksiä. Tarkat kysymykset eivät kuitenkaan aina ole tarkoituksenmukaisia, koska tutkittava ilmiö voi olla laaja ja vastaukset monimuotoisia. Tämän takia painopiste haastattelussa oli oikeiden teema-alueiden sisällä pysymisessä.



**Kuva 7 Teema-alueet tutkimuskokonaisuudessa (mukaillen Hirsjärvi & Hurme 2008, s 67)**

Teema-alueet jaettiin tässä tutkimuksessa kahteen ryhmään, koska palvelun tuottamiseen osallistuvien henkilöiden roolit voitiin pääsääntöisesti jakaa asiantuntijoihin ja palvelujohtamiseen. *Asiantuntijarooleille* haastattelun teemat keskittyivät asioihin, jotka on tunnistettu tiimityön ja tiimin käyttämien menetelmien kannalta keskeisiksi. *Palvelujohtajien* teemat keskittyivät useamman tiimin, asiakasvaatimusten ja työn organisoinnin ympärille. Teemat olivat kuitenkin osittain yhtenevät, koska riippuen haastateltavan roolista,

saattaa henkilöllä olla kummankin roolin vastuita tai kokemusta. Apuna käytettiin miellekarttaa, jossa teemat olivat jaettu vielä aliteemoihin. Miellekartta on kokonaisuudessaan esitetty tämän työn liitteenä.

Asiantuntijaroolin teemat ja niiden valintaperusteet:

- Sidosryhmät
  - Asiantuntijan ja asiantuntijan tiimin tärkeimmät sidosryhmät ovat tietotekniikanpalveluyrityksessä oma johto, asiakkuuden eri henkilöt, yrityksen muut organisaation osat sekä muut saman asiakkuuden tiimit. Tällä teemalla haettiin rajauksia ja ymmärrystä siihen, minne ja miksi asiantuntija tai asiantuntijatiimi kommunikoi ja mitä vaatimuksia näistä kanavista tulee.
- Tiimi
  - Palvelussa oli otettu tutkimusvuoden aikana käyttöön tiimikokoonpanot. Tiimikokoonpanot olivat erilaisia kooltaan, osaamisiltaan ja vastuualueiltaan. Haastatteluun oli valittu jokaisesta tiimistä vähintään yksi kokenut asiantuntija, jolla oli kokemusta ja näkemystä tiimityönsä analysointiin. Teemoitus perustui tiimityöhön ja sen keskeisiin ulottuvuuksiin, jotka tunnistettiin tässä työssä esitellyn asiantuntija- ja tiimityön kirjallisuuden ja tutkimusten pohjalta.
- Tehtäväkenttä
  - Tehtäväkentän teema oli isoin ja laajin, koska työn ydinongelmana oli tunnistaa nimenomaan tehtäväkenttään liittyviä tekijöitä. Kohdepalvelussa tiedettiin asiantuntijatehtävien jakautuvan ohjelmistokehitykseen ja valmistuotteiden konfigurointiin, joiden tehtävienhallinnan luonne poikkeaa toisistaan. Tehtävien virtaukseen ja hallintaan tunnistettiin useita eri vaikuttavia tekijöitä jo läpikäytyjen teorioiden perusteella, joiden pohjalta luotiin aliteemat kysymysten tueksi. Nämä tekijät löydettiin käymällä läpi teoriaa ohjelmistotuotannon ja palvelunhallinnan tunnetuimmista viitekehyksistä sekä tutustumalla Lean-ajatteluun.

Palvelujohdon teemat ja niiden valintaperusteet:

- Edelliset asiantuntijaroolin teemat käytiin myös soveltuvin osin palvelujohdon kanssa riippuen henkilön vastuualueesta.
- Palvelunjohtaminen
  - Palvelunjohtamisen kannalta haettiin näkemystä tutkimusvuoden aikana käyttöönotettuun tiimimalliin ja sen vaikutuksiin työtehtävien hallintaan ja asiantuntijoiden tiimityön johtamiseen. Lisäksi yritettiin löytää taustatekijöitä, jotka vaikuttavat työtehtävien virtaukseen ja -hallintaan.



Yksittäisen haastattelun ajankäyttö suunniteltiin niin, että pääpaino haastattelun teemoissa ja niihin liittyvissä kysymyksissä liittyisi haastateltavan tehtäväkenttään ja siihen vaikuttaviin tekijöihin.

### 3.2.3 Haastattelujen toteutus ja analysointi

Haastattelut suoritettiin tutkimusvuoden loppupuolella (loka-marraskuussa 2018) jolloin muodostetut tiimirakenteet olivat olleet noin puoli vuotta käytössä.

Tiimien henkilömäärät vaihtelivat kolmesta hengestä kahdeksaan. Haastatteluun valittiin jokaisesta tiimistä vähintään yksi asiantuntija. Haastateltava asiantuntija valittiin tunnetun työuran ja -kokemuksen perusteella, jotta haastateltavalla olisi mahdollisimman laaja-alainen näkemys ja kokemus arvioida nykyistä ympäristöä. Palvelun johtamisen rooleihin valittiin palvelupäällikkö ja neljä projektipäällikköä. Haastateltavia oli yhteensä 12 henkilöä, joiden roolijako ja kokemus jakauma esitetty kohdassa Taulukko 3. Palvelussa oli tällöin yhteensä seitsemän eri tiimiä, joista kolme työskentelee ohjelmistokehityksen ja neljä valmistuotteiden konfiguroinnin parissa.

**Taulukko 3 Haastateltujen tiedot**

Haastattelun kohderyhmä	Haastateltavien määrä	Työvuosien keskiarvo	Haastattelun keston keskiarvo
Palvelun asiantuntijat eri tiimeistä	7	12,5	54 min.
Palvelun johtamisen roolit	5	23,4	56 min.

Haastattelutilanteeseen käytettiin 1,5 tunnin aikavarausta jokaista haastateltavaa kohden, josta ensimmäinen vajaa puolituntia tutustumiseen, asian esittelyyn ja loput itse haastatteluun. Haastattelut suoritettiin kahden kesken neuvottelutilassa, jossa ei ollut muita paikallaolijoita. Haastattelujen alussa korostettiin luottamuksellisuutta, jolla varmistettiin, että haastateltava toisi asiat mahdollisimman avoimesti esiin tutkimuksen käyttöön.

Haastattelujen analyysitapoja on kahta tyyppiä: *Selittäminen*, jossa tilastollisesti pyritään selittämään löydetty asiat. *Ymmärtäminen*, jossa käytetään laadullista analyysiä ja päätelmiä (Hirsjärvi et al. 2007). Tämän työn haastatteluissa sovellettiin ymmärtämistä analyysimenetelmänä, koska sillä voidaan selvimmin kytkeä teemat esitettyihin teorioihin ja menetelmiin. Haastattelujen vastaukset kirjoitettiin teemoittain auki, jonka jälkeen litte-roinneista koostettiin teemoihin vastaustiivistelmät. Vastaustiivistelmän pohjalta suunniteltiin kehitystoimenpiteet, jotka perustellaan esitetyllä teoreettisella viitekehyksellä.

## 4. TUTKIMUKSEN TULOKSET

Tämä luku esittelee aluksi tutkimuksen haastattelusta tehdyt vastaustiivistelmät. Vastauksista johdetut kehitystoimenpiteet ja lopuksi keskustelun työn tuloksista.

### 4.1 Tunnistettujen teema-alueiden vastaukset

Tässä alaluvussa on avattu kunkin pääteema-alueen alta löytyneet vastaukset tiivistettynä. Vastaukset on jaettu vielä aliteemoihin, joihin haastattelun kysymykset ja ohjaus pohjautuivat. Teema-alueet aliteemoinen on esitetty listattuna kohdassa Taulukko 4.

*Taulukko 4 Haastattelun pääteemat ja niiden aliteemat*

Haastattelun pääteemat	Aliteemat
<b>Tiimi</b>	Tiimin kokoonpano, muiden yksilöiden mieltäminen tiimiksi ja nykyinen sosiaalisaation aste  Kommunikointi ja lähityö  Tiimin jäsenten roolit ja vastuut  Tiimin yhteinen tavoite
<b>Sidosryhmät</b>	Keskeiset sidosryhmät ja niiden kanssa kommunikointi  Riippuvuudet muihin palvelun tiimeihin
<b>Tehtäväkenttä</b>	Työtehtävätyyppien jakauma  Tiimin teknologiaosaaminen  Työtehtävien tulokanavat  Työtehtävien priorisointi  Työtehtävien volyymin skaalaus  Työtehtävien jaksotus ja synkronointi  Työtehtävien virtaavuus

	Tehtävienhallinnan menetelmät
<b>Palvelujohto</b>	Tiimien johtaminen  Työtehtävien virtaus ja hallinta  Riippuvuudet ja skaalautuvuus tiimien välillä  Liiketoiminnan vaatimustenmukaisuus

Seuraavaksi vastausten keskeinen sisältö on tulkittu kunkin kursivoidun aliteeman jälkeen. Ensiksi esitellään asiantuntijoiden aliteemojen vastaukset, jonka jälkeen palvelujohtoon vastaavat tiivistelmät.

### **Asiantuntijatyön vastaustiivistelmät**

*Tiimin kokoonpano, muiden yksilöiden mieltäminen tiimiksi ja nykyinen sosialisointiaste:* Tiimiin kuuluvuus annetussa organisaatiossa tunnustettiin ja tiedostettiin, mutta tiimiksi koettiin enemmän samaa työtä ja palvelua tekevät, saman paikkakunnan henkilöt kuin tutkimusvuoden alusta luodut tiimirakenteen henkilöt. Henkilöt, joiden työrooliin kuuluu kommunikaatio useamman tiimin jäsenten kanssa, eivät kokeneet vahvaa kuuluvuutta mihinkään tiettyyn tiimiin. Pitkän työuran tehneet, osittain samassa organisaatiosakin, kokivat yhteenkuuluvuutta myös palvelun tai yrityksen ulkopuolella oleviin kollegaryhmiin, joiden kanssa oli toimittu jo vuosia aiemmin pitkissä projekteissa.

*Kommunikointi ja lähiyö:* Eri paikkakunnilla työskentelevät, mutta silti saman tiimin jäsenet, saatettiin kokea etäisemmäksi tai vieraammaksi ellei heidän kanssaan oltu työskennelty pitkiä aikoja vähintään etäyhteydessä.

*Tiimin jäsenten roolit ja vastuut:* Koettiin selkeäksi siltä osin mitä rooleja on jaettu tai osoitettu tiimissä, jos kyseessä oli esimerkiksi Scrum Master, testaaja tai kehittäjäroolien jaottelu tiimitasolla. Valmistuotteiden konfigurointia tekevien roolijaot olivat selkeät. Käytännön töiden jakautumisessa, isoista tuotekohtaisista osaamiseroista johtuen vastauksissa nousi esiin tiettyjen teknologioiden ja tätä myöden henkilöityminen tiimin sisällä, jossa vastuut miellettiin eriävästi tiimin sisällä. Tunnistettiin ettei Scrum Master nimikkeeseen alla tehty puhtaasti Scrum Master rooliin kuuluvaa työtä eikä toisaalta Scrum menetelmänä nimestään huolimatta ollut käytössä.

*Tiimin yhteinen tavoite:* Haastateltavista ketään ei tunnistanut tiimille annetun tavoitteita, joihin tähdättäisiin tiimityön yhteisen tekemisen tasolla. Olemassa olevaksi tavoitteiksi

miellettiin tuotekehityksen vaatimuksissa pysymisen, joustavuuden, työn dokumentoinnin tai palvelutuotannon häiriöttömyyden varmistamisen, muttei näitä koettu viestityn tiimin tavoitteina, vaan koetaan tulevan päivittäisen tekemisen oletetuista vaatimuksista.

*Keskeiset sidosryhmät ja niiden kanssa kommunikointi:* Asiakkaan työjonoihin ja -tehtäviin näkyvyys on parantunut verrattuna aiempaan malliin, jossa työt jaettiin suoraan henkilöille. Töiden suoran jakamisen sijaan osassa tiimeissä siirrytty keskitettyyn tiimin työjonoon (engl. backlog). Työjonosta tiimin jäsen poimii uusia töitä edellisen valmistuttua.

Nyt tiimi tai tiimin nimetty vastuullinen on aktiivisemmin yhteydessä tiimin töiden välittämiseen asiakkaalta tiimille.

*Riippuvuudet muihin palvelun tiimeihin:* Teknisesti tiimien vastuu sovellukset- ja järjestelmät hyvinkin verkottuneita keskenään, mutta arkipäiväisesti kommunikaatio vähäistä muiden tiimien kanssa. Useammassa vastauksessa tiedostettu ja tuotu esiin, että asiakkaan palveluympäristö kehittymässä yhä teknisesti verkottuneempaan suuntaan, jossa tiimin osaaminen voi olla myös siilo, joten tiedon- ja ymmärryksen leviämiseen pitäisi keskittää yhä enemmän panoksia.

*Työtehtävätyyppien jakauma:* Palvelun asiantuntijoiden työtehtävätyypit jakautuivat pääsääntöisesti kahteen alueeseen: Ohjelmistokehittäjät, jotka keskittyivät yhden tai useamman sovelluksen kehitykseen ja ylläpitoon omassa tiimissään. Toinen ryhmä oli valmistuotteiden konfigurointi, jossa järjestelmäasiantuntijoista koostuvat tiimit ylläpitävät tiettyjä yrityssovelluksia ja tekevät niihin kehitystyönä uusia konfigurointeja tai laajennuksia tuotteen ominaisuuksien rajoissa.

*Tiimin teknologiaosaaminen:* Ohjelmistokehitystiimeissä teknologiapinot määrältään vähäisempiä ja yhdenmukaisia. Lähtökohtaisesti näiden tiimien jäsenet osaavat teknologioita tasaisemmin, koska tekevät yhden kehitettävän sovelluksen ympärillä töitä. Tällöin sovellus tyypillisesti koostuu samasta teknologiapinosta. Järjestelmäkehitystiimeissä osaamiserot isoja, koska tiimillä useampi täysin erillinen tuote (valmissovellus) ja tekeminen henkilöitynyttä. Säännönmukaisten tiedon ja työtehtävien jakamisprosessien tai niiden vähäisyys koettiin osasyynä henkilöitymiseen.

*Työtehtävien tulokanavat:* Palvelunhallinnan työt eli häiriö- ja palvelupyynnöt tulevat prosessin mukaan palvelupistesovelluksen (Service Desk) kautta tiimille. Kehitystehtäviä enimmäkseen tiimillä jopa viiden eri projektin kautta. Näiden sovittujen kanavien lisäksi ei koettu merkittävästi ongelmaa epävirallisten tai näkymättömien työkanavien töistä. Työpyyntöjä voi tulla satunnaisesti ohi pääkanavien, esimerkiksi sähköpostin tai kahvipöytäkeskustelujen kautta.

*Työtehtävien priorisointi:* Palveluhallinnan häiriöpyynnöt koettiin selvästi tärkeämmäksi ja korjataan aina kehitystöiden jäädessä toissijaiseksi. Häiriökorjaukset eivät olleet merkittävästi haitanneet kehitystöiden kulkua, koska häiriökorjausten ollessa maksimissaan

keskimäärin muutamien tuntien luokkaa, ei tästä aiheudu useamman päivän kehitystyölle merkittävää keskeytystä. Yksittäisen projektin sisällä olevien töiden prioriteetti koettiin yleensä riittävän selväksi. Epäselvyyksiä koettiin eniten kehitystöiden suhteen, jossa työt kilpailevat rinnakkaisen projektin kanssa jaetusta ajasta. Pahimmillaan kilpailun voittaa eniten tehtävän perään kyselevä asiakas tai projektipäällikkö, eikä yhteisesti hallittu ja sovittu priorisointi, jota ei tunnistettu.

*Työtehtävien volyymin skaalaus:* Tiimirajat ylittävät kehitettävät tai ylläpidettävät yksittäiset sovellukset eivät ole olleet niin laajoja, että tekemiseen olisi osallistunut rinnakkain monestikaan useampi tiimi. Tiimin sisällä kuormituksen vaihtelu on voinut olla merkittävä. Tunnistettiin monessa vastauksessa, että asiakkaan budjetti- ja työnohjauksellisista syistä voivat työkuormat olla alkuvuodesta merkittävät ja toisaalta vuosipuoliskon lopulta merkittävän vähäiset.

Tiimin sisäisesti yksittäisten asiantuntijoiden kohdalla koettiin merkittäviä aika-ajoin tulevia töiden kasautumisia, jotka johtuivat osittain rinnakkaisten töiden tekemisestä samaan aikaan. Rinnakkaisen työn kertymisen yksi toistuvasti mainittu syy oli töissä, jotka otetaan työn alle, mutta suorittamisen aikana työtehtävä jääkin odottamaan tiimin tai asiakkaan toimenpiteitä, eikä sitä voi suunnitellusti edistää aikatauluttaen. Tällöin tekijä ottaa tyypillisesti uuden työn itselleen, jolloin voi syntyä useamman rinnakkaisen työn tilanne, jotka purkautuvatkin samalla hetkellä aiheuttaen ylikuormaa.

*Työtehtävien jaksoitus ja synkronointi:* Joissain kehitystehtävissä on tarvittu useamman tiimin yhtäaikaista kehitystä ja julkaisua. Näissä ei ole käytetty mitään valmista viitekehystä tai menetelmää, vaan työt on lähinnä koordinoitu projektinhallinnan toimesta ja yksittäisten eri tiimien asiantuntijoiden kommunikointiin.

*Työtehtävien virtaavuus:* Tutkimusvuoden aikana jokaisella tiimille muodostunut tiimin työjono yksilöllisen töiden jakamisen sijaan. Töiden sisältö työjonossa kehityksen aloittamiseen hyvin vaihteleva. Työtehtävien kuvaukset toisinaan liian yksinkertaistettuja tai abstraktiotaso korkea, jolloin kehittäjä ei kykene etenemään ilman tarkentavaa määrittelyä.

*Tehtävienhallinnan menetelmät:* Ohjelmistokehitystiimeissä siirrytty vesiputousmallista kohti ketterämpiä menetelmiä tutkimusvuoden aikana. Osassa tiimeissä työt jaettiin henkilöohjatusti. Etukäteispainotteisesta suunnittelusta siirrytty aluksi kohti Scrum-viitekehystä, josta tällä hetkellä lähemmäs Scrumban-tyyppistä kehitystä. Järjestelmäkehitystiimeissä enimmäkseen käytössä annettuna Scrum-mallin jalkautus.

## **Palvelujohtamisen vastaustiivistelmät**

*Tiimien johtaminen:* Johtamisen vastuualueet jakautuivat palvelutuotannon johtamiseen palvelupäällikön roolilla ja kehitystyön johtamiseen useiden eri projektipäälliköiden rooleilla. Tiimityöstä ketterässä kehitysmallissa tai ketterän työn kontekstissa oli haastatelluilla johtajilla hyvin vähän kokemusta.

Tutkimusvuoden aikana oli siirrytty yksilöpohjaisesta työtehtävien jaosta kohti tiimipohjaista mallia. Tämän jälkeen lähes kaikki johtajaroolilliset kokivat töiden jakamisen hallinnan parantuneen tai ainakin pitivät muutosta neutraalina. Haasteita koettiin vielä tilanteessa, jossa useamman projektipäällikön töitä suoritetaan rinnakkain samassa tiimissä. Tällöin oman projektin töiden eteneminen ei ollut varmuudella hallinnassa, joka toi vaikeuksia oman projektin edistämiseen ja hallintaan. Priorisoinnista ja henkilöiden kapasiteetin jaosta oli pidetty resursointipalavereja, mutta käytännössä niissä sovittu toteutuma on koettu epävarmaksi nopeasti muuttuvien ja osittain tuntemattomien kapasiteetinvaihtelujen takia.

*Työtehtävien virtaus ja hallinta:* Työtehtävien virtauspoluissa (”arvoketjussa”) asiakkaan liiketoiminnalta asiantuntijoille oli projekti- ja liiketoimintokohtaisia suuria eroja. Palvelun sovelluksia ylläpidetään asiakkaan usealle eri liiketoiminnalle. Riippuen asiakkaan liiketoiminnan henkilöistä, esimerkiksi osaamistasot ketterään tekemiseen voivat vaihdella suuresti. Asiakkaan liiketoiminnan vastuuhenkilöiden ollessa pitkään samat, oli muodostunut selkeästi ketterän kehityksen elementtejä, kuten tuoteomistajavastuullinen ja ymmärrys tuotteen iteratiivisesta kehittämisestä. Toisessa ääripäässä liiketoiminnalla oli vain sovelluskohtainen ”toiveiden tynnyri” eikä mielenkiintoa osallistua budjetäärisiin päätöksiin, töiden priorisointiin tai arkkitehtuuriin vaikka ne määrittävät vahvasti toimitajan tekemistä.

Ohjausryhmät eivät olleet kaivanneet tiimin tai tehtävien virtauksen statistiikkaa tai metriikkaa. Palvelutasoa ja häiriöhallintaa seurattiin säännöllisesti.

*Riippuvuudet ja skaalautuvuus tiimien välillä:* Vastauksissa tunnistettiin, että toiminnallisuudet eri ylläpidettävissä järjestelmissä ja sovelluksissa ylittävät tiimirajat. Tämän monimutkaisuuden lisääntyminen todettiin mahdollisesti aiheuttavan lisää ongelmia tulevaisuudessa ja sen hallitsemiseen pitää kiinnittää huomiota.

Kehitystoissa vähemmistöä oli kuitenkin työt, jossa työt pitäisi ajoittaa usean tiimin kesken yhtä aikaa suoritettavaksi tai skaalautuisi yli tiimirajojen.

*Liiketoiminnan vaatimustenmukaisuus:* Sopimustason palvelutasovaatimuksien lisäksi laadullisia määriteltyjä mittareita ei juurikaan tunnistettu. Pidettiin tärkeänä, että asiantuntijat kuitenkin ymmärtävät laadun vastuun.

Asiakkaan budjetoinnin ennustamisen vaikeus ja näkymättömyys, joka heijastuu haasteena töiden jakamiseen ja suunnitteluun. Tämä nousi isoimpana koettuna syynä kuormi-

tuksen vaihteluun tiimeissä. Myös osaaminen jakautunut tiimeissä epätasaisesti. Henkilöitynyt osaaminen tai epäselvyys osaamisesta nousi lähes kaikilla yhdeksi haasteeksi tiimin tasapainoiselle jakautumiselle tiimin jäsenten kesken.

Asiakkaan työtilausmalli toimittajalle koettiin esteeksi aitoon tiimipohjaiseen kaistapohjaiseen laskutukseen siirtymiseen. Nykymallissa työt niputetaan aina yhdeksi seurattavaksi kokonaisuudeksi, joka sisältää arvioidun ja sovitun työmäärän, jonka toteuma on laskutusperuste. Tähän malliin tiimin koko kapasiteetin myymisen ja ketterän toiminnallisuus iteroinnin sovittaminen koettiin lähes mahdottomaksi.

## 4.2 Tehtävienhallinnan käyttöönotto ja kehitys yrityksessä

Tässä alaluvussa kuvataan välittömät toimenpiteet, joita pystyttiin tekemään tutkimuksessa tunnistettujen tulosten pohjalta. Toimenpiteet on perusteltu pohjautuen aiemmin työssä esiteltyyn kirjallisuus- ja tutkimustietoon.

### Tehtävienhallinnan menetelmiä käyttöönottoa tukevat tekijät

Tiimin toiminta ja yksilön kokemus on perusta tiimin toimivuuden ja yhteenkuuluvuuden kannalta. Tutkimushetkellä kohdepalvelun käyttämä tiimimalli oli vasta käyttöönotettu ja vastauksissa ilmenikin, että nykyiseen nimettyyn tiimiin kuulumisen ei ollut vielä vahva. Sosialisaatiota koettiin enemmän entisiin, tai lähikollegoihin maantieteellisesti hajautetun tiimin tapauksessa. Koska sosialisaatioprosessin todettiin kirjallisuudessaakin olevan tärkeä, tulisi muodostettujen tiimien olemassa oloa tukea ja vahvistaa pitkäjänteisesti kuten alaluvussa 2.3.3 avataan.

Kirjallisuudenkin mukaan oikean tiimikoon määrittäminen on kompleksinen ongelma kuten alaluvussa 2.3.2 esitetään. Verrattuna esimerkiksi 5-8 henkilön tiimeihin, palvelun kolmen hengen tiimit olivat osaamiseltaan henkilöityneet eikä tiimityön menetelmiä oltu juurikaan mietitty, joten henkilöiden uudelleenorganisointia laajempiin tiimeihin kannattaa kehittää lisää, jotta päästäisiin saavuttamaan enemmän tiimityön ja tiimien tehtävienhallinnan hyötyjä.

Tiimin yhteenkuuluvuutta lisäävänä tekijänä tiimille tuli löytää käytännön palvelutuotannon- ja kehitystöihin liittyviä yhteisiä tavoitteita. Tavoitteet voidaan esimerkiksi löytää tiimin päivittäisten työprosessien kehittämisen tai suorittamisen ympäriltä, jolloin niiden onnistuvuuden mittaaminen on mahdollisimman konkreettista. Yhteiset tiimin tavoitteet tukevat myös muokkautumista itseohjautuvaksi tiimiksi, tiimin elinkaaren kehitystä ja täten parantavat suoritustaitoa kuten alaluvussa 2.3.4 ja 2.3.5 esitettiin. Tiimiä tulee siis johtaa ja tukea niin, että se kykenee muodostamaan vahvan sosialisaation jäsenten kesken ja saavuttamaan tiiminä suorittavan vaiheen. Tämän takia palvelujohtoon asettamien tiimitavoitteiden asetus on tärkeää. Alaluvussa 2.5.2. esitetty ketteryyden tunnettu julistus tuo periaatteissaan esiin myös tiimin ja motivoituneet yksilöt.

Tiimityöstä ja sen hyödyistä oli haastateltavilla johtajilla vain vähän kokemusta. Palvelussa aloitettiin työprosessien kehittäminen ja kuvaaminen, sekä ohjaaminen kohti itseohjautuvampia tiimejä keskitetyn johtamisen sijaan. Nämä ovat erittäin tärkeitä tiimin johtajuuden muutoksia kuten alaluvuissa 2.2 ja 2.3.4 esitetään, jotta parannettaisiin tiimien vaikuttavuutta ja yksilöiden motivaatioita. Toinen kirjallisuudessa mainittu tärkeä kategoria on tiimin suorituskyvyn ja vaikuttavuuden hallintaa. Tähän kuuluu yhtenä osa-alueena opettaa tiimin asiantuntijoille tiimissä toimisen taitoja ja kulttuuria. Haastattelujen myötä tunnistettiin, että tiimitoiminnan fasilitointia on tutkittavassa palvelussa liian vähän, ja sen osaamista pitää kehittää asiantuntija- ja johtajatasolla. Nostettiin esiin, että moni tiimi oli fyysisesti jakautunut usealle eri paikkakunnalle, jolloin kirjallisuuden mukaan on tärkeää levittää entisestään johtamista ja itsenäistä päätöksentekoa tiimille ja sen yksilöille. Keskitetty vahva johtajuus heikentää suorituskyyä laajalle levinneissä tiimeissä. Myös itseohjautuvien eli autonomisten tiimien myötä johtajuus muuttuu, joten tämä vaatii johtajilta oppimista uuteen rooliin.

### **Tehtävienhallinnan menetelmien käyttöönotto ja ominaisuudet**

Palvelun liiketoimintamallissa henkilöresurssi oli laskuttava yksikkö, jolloin lyhyen tähtäimen tavoite oli varmistaa henkilön maksimaalinen käyttö, josta seuraisi maksimaalinen laskutus. Kuten alaluvussa 2.4.1 esitetään, on virtaustehokkuuden korostaminen lähtökohtaisesti asiakas-arvoa tuottavampaa resurssitehokkuuden sijaan. Työtehtävien tehokkuuden ja arvontuoton kannalta tulisi kuitenkin keskittyä ensisijaisesti virtaavuuteen, ja vasta sen jälkeen mitoittaa resurssit oikealle tasolle käyttöasteen, esimerkiksi hyvän ja kannattavan laskutuksen varmistamiseksi. Palvelun haastatteluissa tuli esiin, että aiemmin ennen tiimimallin kehittämistä, oli keskitetty täysin henkilö- eli resurssiohjattuun työn jakoon, jossa havaittiinkin esiin alaluvussa 2.4.1 esitettyjä ongelmia resurssitehokkaaksi ohjautuneessa palvelussa.

Virtauksen visualisoinnin lisäksi on olennaista jatkossa kuvata kaikki työtehtävien virtauskanavat ja niiden prosessit, jotta asiantuntijalla on selkeä kuva työn tulo- ja toimituskanavista, sekä johdolla parempi kyvykkyys nähdä ja mitata työtehtävien virtaa. Johtamisen ja suorituskyvyn nostaminen edellyttää aina oikeata mitattavuutta, jotta saadaan takaisinkytkentä tehtyihin muutoksiin ja niiden toimivuuteen.

Järjestelmäkehitystä ja palvelutuotantoa tekevissä tiimeissä otettiin käyttöön menetelmät, jossa työtehtävien virtausprosessi visualisoidaan ja päiväkokouksessa liikutellaan tehtäväkortteja edistymisen mukaan. Tämä tehtiin johdettuna Kanban-menetelmän ensimmäisen vaiheen eli prosessin visualisoinnin perusteella. Tämän koettiin parantavan välittömästi tiimin näkyvyyttä töihin ja auttoi kaikkia tiimin jäseniä ja osapuolia ymmärtämään, miten työt virtaavat. Tehtäväkorttien avulla saatiin rinnakkainen työ näkyviin, jota asiantuntijoilla saattoi olla käynnissä useampia. Tehtäväkortteille luotiin myös tilat, jolloin paljastui pullonkauloja esimerkiksi tietyn asiakkaan liiketoimintojen osalta. Kuten alalu-



vussa 2.6.2 esitetään, nämä toimenpiteet ovat olennaisia Kanban-menetelmää käyttöönotettaessa. Tehtävavirran visualisoinnin jälkeen tulisi tiimien löytää oikea arvo keskeneräisen työn rajoittamiseen ja käyttöönottaa se prosessiin. Tällöin tiimi oppii ymmärtämään ja rajoittamaan omaa keskeneräisen työn määrää ja löytämään tarkemmin oman virtauskyvykkyytensä eri töille. Keskeneräisen työn rajoittimen lisäksi tulisi vaihteittain lisätä prosessin mittausta ja hallintaa.

Havaittiin, että ohjelmistokehitystä ja palvelutuotantoa rinnakkain tekevä tiimi ei voi yksistään ja täysin tukeutua pelkkään Scrum-malliin, koska palvelutuotannon häiriöhallintatyön vaihtelu voi olla suurta. Toissijaisesti myöskään rinnakkaiset projektien työt eivät tahdistu keskenään, jolloin Scrum Sprintin suunnittelu on lähes mahdotonta ja eikä tarkoituksenmukaista. Täten ei ollut mielekästä lukittua tietyn Scrum-mallin sprintin sisältöön kiinteästi, kuten Scrum-viitekehityksessä toimitaan. Palvelussa oli kaksi enimmäkseen tuotekehitystä tekevää tiimiä, mutta näissäkin tiimeissä luovuttiin puhtaan Scrum-menetelmän käytöstä ja siirryttiin osittain Kanban- ja Scrum-mallin sekoitukseen, koska rinnakkain tehtiin eri palvelutason tehtäviä ja toisaalta kehitettiin samalla tiimillä kahta tai useampaa sovellusta. Oli myös mielekästä, että Sprintit eivät määrittele kiinteää sisältöä, koska mitään tiettyä tavoitetta sprintille ei tunnistettu. Asiantuntija voi ottaa priorisoidusta tiimin työjonosta tärkeimmän kehitystyötehtävän työn alle. Tällöin tiimistä voi virrata useamman eri sovellustuotteenkin arvoa tuottavia inkrementtejä toisistaan riippumatta.

Jokaiselle tiimille luotiin yksi keskitetty tiimin tehtävälista, johon kaikki asiakastyöt eri projekteista virtasi. Tämä oli lähinnä Scrum-mallin mukaista tuotteen tehtävälistaa (engl. product backlog), mutta tehtävät koostuivat usean eri järjestelmän- tai tuotteen kehitystehtävistä, toisinkuin Scrum-mallissa kehitys tyypillisesti kohdistuu yhteen tuotteeseen.

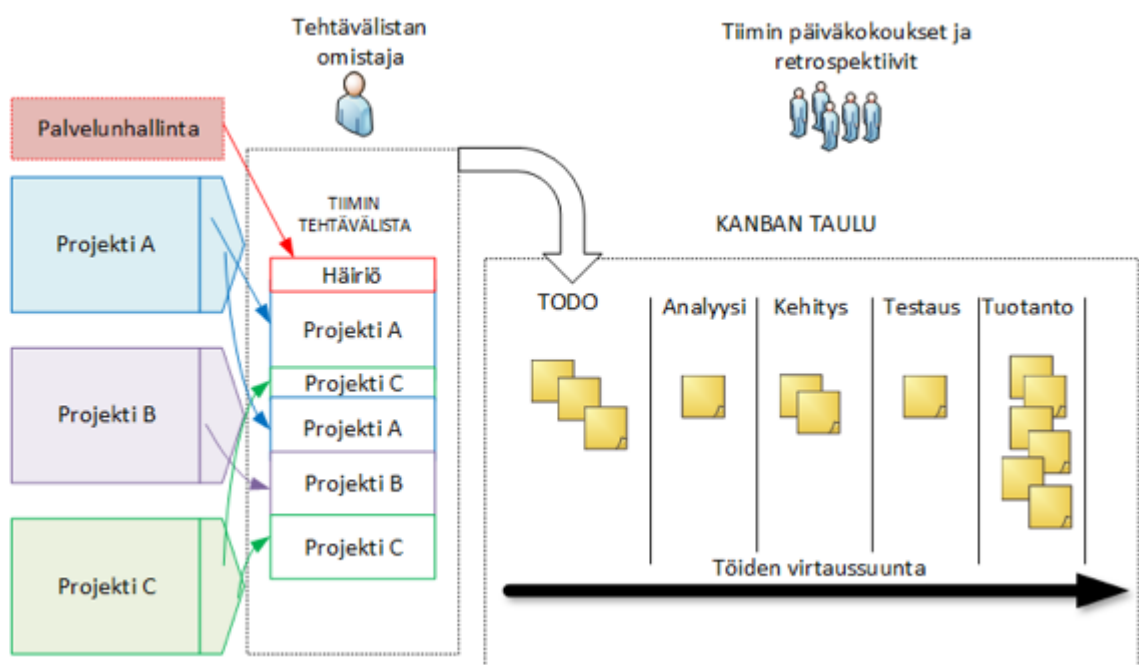
Tehtävälistan jalostuskokouksia aletaan tehdä tarvittaessa, kun ymmärretään projektikohdainen raja, jossa ei enää ole riittävästi tehtäviä aloitettavaksi. Jalostuskokouksia ei kannata tehdä toistuvasti tietyn ajan välein, vaan määrittää tarve virtauksen mukaan kuten Scrumban menetelmässä sovelletaan alaluvussa 2.6.2. Liian tarkka määrittely liian pitkälle voi luoda hukkaa, toisaalta virtauksen varmistamiseksi, pitää jalostuskokous pitää riittävän usein.

Luotua tehtävälistaa priorisoitiin säännöllisesti niin, että kaikki tiimin työtehtäväkanavien työt olivat rinnakkain kanavien keskinäisen kiireellisyys- tai palvelutason mukaan. Tehtäväkanavakohtaiset tehtävät olivat taas keskinäisessä prioriteettijärjestyksessä sen mukaan mitkä tehtävät olivat kullekin kanavalle tärkeimpiä. Tällöin tiimin henkilön ottaessa uutta työtä, saa hän priorisoinnin ansiosta sen työn, joka on kanavakohtaisesti tärkein. Tämän lisäksi myös eri työkanavalle toteutuu tehtävien valmistumisen kiertoa, koska listassa on eri tehtäväkanavien tehtäviä. Tällä vältettiin erityisesti se, ettei tiimi keskity te-

kemään yhtä sovellusta tai tehtäväkokonaisuutta estäen muiden kanavien töiden valmistumisen. Priorisointiin, rinnakkaisen työn rajoihin ja tiimin tehtävälisan luontiin sovellettiin Kanban-kirjallisuudessa esitettyjä eri vaiheita, jotka kuvattiin alaluvussa 2.6.2.

Kanban-menetelmä itsessään ei määritellyt tiimin tai työtehtävien suorittamisen rooleja tai vakionuotoisia kokouksia, vaan on keskittynyt prosessin visualisointiin, tutkimiseen ja kehittämiseen. Kuten Lean alaluvussa 2.4 esitetään, on yksi tärkeä elementti toiminnan jatkuva parantaminen. Scrum-mallia mukailleen otettiin käyttöön tiimin retrospektiivikokoukset, jotta säännöllisin väliajoin voidaan tarkastella nykyistä prosessia ja käytettyjä toimintamalleja, sekä kehittää niitä edelleen. Prosessin ja tiimin oman toiminnan tarkastelu korostuu myös ketterän julistuksen periaatteissa alaluvussa 2.5.2.

Lisäksi luotiin tiimeille päivittäiskokoukset, joissa seurataan töiden edistymistä ja taasaista jakautumista tiimin henkilöille, sekä tuodaan esiin koettuja ongelmia tiimin ratkaistavaksi. Kuva 8 visualisoi tämän alaluvun keskeiset tehtävähallinnan kehityskohteita ja toisaalta kohti tavoitetilaa. Tehokkaasti toimiva tiimi järjestäytyy päiväkokoouksissa Kanban-työkalun ääreen, jossa käydään läpi tehtävien kulku ja niihin liittyvät mahdolliset ongelmat. Retrospektiivit järjestetään säännöllisesti toiminnan parantamiseksi. Projektien keskinäiset prioriteetit, sekä projektin sisäiset prioriteetit määräävät tiimin tehtävälisan järjestyksen. Tiimin tehtävälisaa hallinnoi vastuurooli, joka yhdistää eri projektien tarpeet. Tiimistä löytyy valmentajarooli, joka auttaa tiimiä menetelmien tuntemuksessa ja koordinoi tehtävien virtaavuutta yhdessä tehtävälisan omistajan kanssa.



**Kuva 8 Tiimin keskeiset tehtävienhallinnan elementit kuvattuna**

Kuten alaluvussa 2.7 onkin yhteen vedetty, Lean-ajattelun ja ketterien menetelmien yhtenäisestä työkalupakista voidaan löytää tiimille keskeisimmät menetelmät ja kuten aitoon ketterää ajatteluun kuuluukin, tiimit valitsevat juuri heidän kontekstiinsa sopivat menetelmät.

### 4.3 Keskustelua työn tuloksista

Tutkimuksen empiirisen osuuden suuntautuessa yrityksen tiettyyn aiemmin tutkimattomaan kohdepalveluun ei vastaavaa täysin rinnastettavaa tutkimustietoa ollut aiemmin käytettävissä. Kuten alaluvussa 2.7 todettiin, on jokainen ympäristö lopulta yksilöllinen kun aletaan soveltaa Leanin ja ketteryuden periaatteita. Jotta sisältöä voitaisiin arvioida muuhun tutkimustulokseen, yritettiin etsiä muita vastaavia tapaustutkimuksia tai kirjallisuuskatsauksia aiheesta.

Kanbanin käyttöä ohjelmistotuotantoon onkin alettu tutkia 2010-luvulla yhä enemmän. Lähes poikkeuksetta sen käyttöönottoon liittyvät tapaustutkimukset eri yhteyksissä ovat korostaneet sen etuja, joita tässäkin tutkimuksessa havaittiin.

Yhtenäisiä havaintoja tähän tutkimukseen oli Marko Ikosen Kanbanin vaikutusta projekteihin tutkivassa väitöskirjassa. Kanbanin eduiksi käyttöönoton myötä saavutettiin: *Prosessista hukan paljastamista, kuten keskeneräistä työtä tai töiden välillä vaihtamista* ja *Hallittavuuden paraneminen tehtävävirran visualisoinnilla* (Ikosen 2011). Kohdepalvelussa käyttöönotetut yhä enemmän tiimien autonomiaa korostavat käytännöt tukivat myös näkemystä jaetun johtajuuden ja yhteisten tavoitteiden tärkeyttä. Näiden todellista onnistuvuutta ja mitattavuutta ei kuitenkaan voinut ajallisesti näin lyhyessä tutkimuksessa luotettavasti havainnoida.

Tapaustutkimuksessa Nikitina et al. (2012) prosessitransformaatio Scrumista Scumbaniin oli keskeinen löytö se, että vaikka Scrumin muokkaaminen Kanbaniksi paransi kohdeyrityksen toimintaa, oli lopulta yhteenvetona se, että prosessi ja organisaatio-ongelmien ratkaisemiseen pitää olla mekanismit jatkuvaan parantamiseen sekä hyvin koulutettu motivoitunut henkilöstö. Retrospektiivien käyttöönotto kohdepalvelussa oli linjassa myös viitatus tapaustutkimuksen kanssa, eli säännöllisellä jatkuvan parantamisen kulttuurilla voidaan saada palautetta ja reagoida nopeasti epäkohtiin.

Kanbanin käyttöönotosta suurelle hollantilaiselle ASR vakuutusyhtiön tietotekniikkaosastolle on myös tehty raportti. Vakuutusyhtiön ohjelmistotuotannossa ja -ylläpidossa yhteensä seitsemälle tiimille ja 200 hengelle oli käynnissä muutosprosessi Kanban-malliin. Yhteistä tähän tutkimukseen oli, että tutkimuksessa vakuutusyhtiössä oli myös tiettyihin teknologioihin sidottuja tiimejä, jotka joutuivat kehitystyön ohella tekemään ylläpityötä. Tätä työtä oli usein mahdoton sovittaa kahden viikon Sprinteille, koska muutokset haluttiin joustavammin ja tiheimmin tuotantoon. Kanbanin käyttöönotto oli onnistunut ja muutoksesta oli saatu positiivisia kokemuksia (Maassen & Sonneveld 2010).

Laajemmat kirjallisuuskatsaukset Kanbanin käytöstä vahvistavat myös sitä, että oikealla käytöllä Kanban vähentää kesken eräisen työn määrää, tuo esiin rajoitteet ja koordinoi tiimityötä. Nämä samat tekijät voitiin havaita myös tässä tutkittavassa kohdepalvelussa siltä osin mitä havainnointia välittömien muutosten osalta tehtiin (Wang et al. 2012; Al-Baik & Miller 2015).

Ohjelmistoyrityksille suunnatun ketteryuden kyselytutkimuksen mukaan 14% vastaajista, kyselyn toiseksi isoin joukko, kertoi käyttävänsä menetelmien hybridiä eikä vain yksittäistä menetelmää. Sama tutkimus vertaili myös menetelmien käytetyimpiä yksittäisiä ominaisuuksia. Näistä kolme käytetyintä olivat: Päivittäiskokoukset, sprintin tai iteraation suunnittelu sekä retrospektiivit (VersionOne 2018). Myös kehitettävän palvelun osalta päädyttiin poimimaan ominaisuuksia eri menetelmistä.

## 5. YHTEENVETO

### 5.1 Keskeiset tulokset

Tietotekniikan palvelualalla yhä useampi yritys pyrkii yksittäisten ohjelmistojen tuotekehitysprojektien sijaan tuottamaan pitkäjänteisempää palvelua, johon kuuluu jatkuvat sovelluksen kehitys- ja ylläpitopalvelut. Tällöin yritys pyrkii organisoimaan tuotettavan palvelun ja valitsemaan siihen oikeat tehtävienhallinnan menetelmät varmistaakseen tehokkaan palvelutuotannon. Tällöin voi organisoituminen johtaa malliin, jossa sama asiantuntijatiimi tekee aikataulutetun kehitystyön lisäksi yllätyksellistä ja kiireellistä tuotannon häiriökorjausta. Tämän diplomityön tarkoituksena oli tutkia tietotekniikan palveluyrityksen tiettyä asiakkaalleen tuottamaa palvelua, jossa asiantuntijatiimit tekevät edellä kuvattua palveluhallintaa- ja kehitystä rinnakkain.

Tarkoituksena oli löytää tehtävienhallinnan menetelmien keskeiset tekijät ja kehittää niiden avulla yrityksen tietyn tutkittavan palvelun palvelutuotantoa. Tutkimuksen ongelma, johon haettiin ratkaisua, oli tiivistetty seuraavaan tutkimusongelmaan:

***Miten valita sovelluskehitystä ja -ylläpitoa tekeville asiantuntijatiimille menetelmät tehtävienhallintaan?***

Tähän tutkimuskysymykseen vastattiin kahden alikysymyksen kautta seuraavasti:

***Mitkä tekijät vaikuttavat tehtävienhallinnan menetelmien valintaan?***

Koska tiimi luo perustan palvelutuotannolle sekä on menetelmien käytön edellytys, on vaikuttavana tekijänä olennaisesti toimiva tiimi. Toimivan tiimin keskeisinä tekijöinä tunnistettiin: *tiimin muodostus, tiimille annettavat yhteiset tavoitteet sekä jaettu johtajuus*. Nämä vahvistavat tiimityön ja yksilön sosialisatioprosessia, joka on pohja menetelmien käyttöön.

*Tiimikohtaisia tehtävävirtoja oli palvelussa tiimillä aina vähintään kaksi, kiireelliset tuotantohäiriöt ja erilliset aikataulutetut kehitystyöt, joten tehtävävirtojen määrä ja niiden toisistaan poikkeavat palvelutasoluokat määrittivät keskeisesti valintaa menetelmien välillä.* Tämän takia otettiin käyttöön tehtävien virtausta tukeva Kanban-menetelmä, koska esimerkiksi Scrum-menetelmässä olisi pakotettu käyttämään kiinteä sisältöistä aikaikkunaa eli Scrum sprinttiä. Kiinteä sisältöinen aikaikkunoitu Scrum sprintti ei sovellu tiimille, joka voi saada yllättäen tuotantohäiriön korjattavaksi. Lisäksi tiimi voi tuottaa useita erillisiä arvoja tuottavia kehitystöitä eri sovelluksiin. Näitä töitä ei ollut tarvetta paketoita yhteen Sprintin inkrementtiin tai julkaista yhdellä kertaa. Palvelussa arvoa luovia inkrementtejä saatettiin julkaista vaikka päivittäin. Useammalle työtehtävävirrälle oli myös ominaista

yksilöllinen tehtävavirtakohtainen priorisointi, joka ei sovellu suoraan Scrum-menetelmään, jossa yksittäinen tuoteomistajarooli vastaa kaikkien tiimin töiden tärkeysjärjestyksestä. Scrum-menetelmän tuoteomistajan kaltainen rooli tarvitaan kuitenkin linkittämään tiimin tehtävälusta eri projektien tehtävavirtoihin.

### ***Mitkä tehtävienhallinnan menetelmien ominaisuudet valittiin käyttöön?***

Keskeisimmät ominaisuudet löytyivät tehtävavirtojen hallintaan Kanban-mallista:

- *Tehtäväkorttien käyttöönotto:* Jokaisella tehtävällä on oma kortti, josta selviävät tekijä, työhön liittyvä projektikokonaisuus (tehtävavirta) ja kiireellisyys.
- *Virtauksen visualisointi:* Tuotetaan näkyvyys työtehtävavirtoihin tiimin ja palvelujohdon tarpeisiin
- *Tiimin tehtävälusta* sisältää tiimin priorisoidun tehtävälistan, johon on yhdistetty eri projektien tehtävavirratt.
- *Tehtävälistan aikataulut, priorisointi- ja palvelutasoluokitus:* Tiimin tehtävavirtojen suunnittelu kiireellisille tehtäville (tuotantohäiriöt) ja aikataulutetuille kehitystehtäville.

Roolit ja kokoukset mukaillen Scrumin ja muiden ketterien menetelmien malleja:

- *Tehtävälistan omistajan rooli:* Tiimillä on Scrum-menetelmän tuoteomistajan kaltainen rooli, jonka tehtävä on koordinoida liiketoiminnan eri projektien tehtävät tiimin tehtävälustalle priorisoituun järjestykseen.
- *Tiimin valmentajan rooli:* Scrum Masterin kaltainen rooli, jolla valmennetaan tiimiä menetelmien käyttöön ja varmistetaan, että tehtävälistan omistajan koordinoimat tavoitteet toteutuvat. Tehtävien saadessa riippuvuuksia muiden tiimien tehtäviin, hoidetaan koordinointi yhdessä *Tehtävälistan omistajien* kanssa.
- *Tehtävälistan jalostuskokoukseen* kokoonnutaan tarvittaessa projektikohtaisesti määritellyn rajan alitettua. Raja mittaa tehtävissä olevien jalostettujen tehtävien määrää. Tiimi määrittää rajan niin, että tiimillä on sopivasti jalostettuja tehtäviä aina jonossa. Jalostuskokouksessa projektin tärkeimpiä tehtäväkokonaisuuksia jaetaan hienojakoisemmiksi tehtäviksi, jotta ne ovat valmiimpia toteutettavaksi.
- *Päiväkokoukset:* Tehtävavirran edistämiseksi kokoonnutaan lyhyeen päiväkokoukseen Kanban-aulun äärelle, jolla varmistetaan töiden eteneminen ja haetaan ratkaisut mahdollisiin

töitä edistäviin ongelmiin. Kokous on lyhyt Scrumin päiväkokouksen kaltainen katsaus menneisiin ja tuleviin tehtäviin.

- *Retrospektiivi* toistuvaiskokouksena tiimityön jatkuvan parantamisen mahdollistaja. Retrospektiivi ei varsinaisesti kuulu mihinkään yksittäiseen malliin tai menetelmään, joskin sen voidaan ajatella olevan vähintään ainakin Scrumin ominaisuus tai osakokonaisuus. Retrospektiivi voidaan nähdä osana tehtävienhallintaa, koska tiimi kehittää sillä omaa tehtävienhallinta prosessiaan, vaikka retrospektiivi tilaisuudessa voidaan käsitellä myös muita tiimityöhön liittyviä asioita.

Edellä mainittujen alikysymysten vastausten yksityiskohtaisemmat tulokset sekä niiden perustelut kirjallisuuteen- ja tutkimustietoon verraten on esitetty alaluvussa 4.2.

## 5.2 Suositukset käytäntöön

Tutkittavan kohdepalvelun käyttöön tutkimuksessa löydetty suositukset ja jatkotoimenpiteet tulee käytännössä perustua aluksi osaamisen vahvistamiseen ja tiimin rakenteisiin, jotta palvelujohdolle ja asiantuntijoille saadaan parempi kyvykkyys analysoida ja kehittää omaa toimintaansa. Osaamisen syventämisen myötä voidaan painopistettä siirtää yhä enemmän tehtävienhallinnan analysointiin ja jatkokehittämiseen. Tämä alaluku esittelee lyhyesti ensiksi *osaamisen ja tiimityön* suositukset, jonka jälkeen itse tiimikohtaiset tekijät *tehtävienhallinnan* kehittämisen. Tutkimuksen aikana tehtiin jo osaltaan välittömiä muutoksia, jotka kuvattiin alaluvussa 4.2.

Osaamisen vahvistamisen ja tiimityön välittömät suositukset asiantuntija ja johtamisen rooleille:

- Jatkuvat retrospektiivit välittömästi kaikkeen toiminnan arkeen, jossa tiimityön kaltaisesti tehdään toistuvia aktiviteetteja samalla henkilökokoonpanolla.
- Tarkistetaan tiimien kokoonpanot ja rakenteet
  - Tiimien henkilöpohja riittävän laajaksi, jotta tiimityön hyötyjä voidaan täysimääräisesti saavuttaa. Henkilöpohjan laajennus vähentää henkilöriskiä, erityisesti osaamista laajentamalla. Laajemmilla tiimipohjilla voidaan myös vähentää hukkaa, joka seuraa kysynnän vaihtelusta eri osa-alueelle. Laajempi ja pysyvämpi pohja tukee myös tiimin sosiaalisuusioprosessin käynnistymistä ja vahvistumista, joka taas tukee siirtymää yksilötyöstä kohti tiimityötä. Henkilöpohjan laajentamisessa pyritään kasvattamaan tiimiä heterogeenisemmäksi. Tiimien monimuotoisuuden pohjaa voi lisäksi vahvistaa tiimityön menetelmiä valmentavilla rooleilla.
  - Määritellään tarkemmin asiakas- ja palvelustrategian tavoitteiden kautta konkreettiset tiimitason tavoitteet. Tiimin yhteinen jaettu tavoite

ja missio vahvistavat tiimin olemassaoloa ja luovat sille olemassaolon tarkoituksen.

- Pyritään keskittämään työvirrat samasta teknologia- ja osaamisalueelta niistä vastuussa olevalle tiimille. Vältetään tekemästä erillisiä projektiryhmiä tiimin resursseista. Sen sijaan ohjataan projektitekemisen tehtävienhallinta osaksi tiimin päivittäistoimintaa muiden töiden rinnalle.
- Syvennetään erityisesti ketterän ja Lean-ajattelun osaamista, sekä tehtävienhallinnan osalta erityisesti Kanban-menetelmän käyttöä ja mahdollisuuksia.

Tiimikohtaiset tehtävienhallintaan liittyvät kehityssuositukset esitetään seuraavaksi. Näiden edellytyksenä on tehtävienhallinnan menetelmien ja periaatteiden riittävä osaaminen tiimissä, jotta tiimi itsessään oppii kehittämään omaa prosessiaan:

- Tiimikohtaisten työvirtojen visualisoinnin parannus ja varmistus, että kaikki työtehtävävirrat ovat aina näkyvillä. Tällöin saadaan todenmukainen kuva tiimin tekemistä töistä ja tehtävien läpimenosta.
- Päivittäiskokousten ja tehtävälistojen jalostuskokousten käyttöönotto tiimin omaan prosessiin soveltaen
- Tiimikohtaisen tehtävävirran tarkempi analysointi säännöllisesti, jotta tiimi kykenee asettamaan ja tarvittaessa muuttamaan keskeneräisen työn rajoitinta.
- Tiimin tehtävävirran läpimenoajan ollessa ymmärretty, kehitystyön aikataulutuksen tukeutuminen tiimin todellisen tehtävävirran nopeuteen yhä vahvemmin.
- Usean tehtävävirtauskanavan keskinäisten priorisointimenetelmien käyttöönotto laajemmin ja näkyvämmiin tiimin tehtävälistoilla. Tehtäväomistajan rooli, joka vastaa tiimin tehtävälistan oikeasta priorisoinnista eri tehtävävirtauskanaviin liittyen.

### 5.3 Työn arviointi ja rajoitteet

Akateemisessa tutkimuksessa on syytä arvioida sen luotettavuutta ja pätevyyttä. *Reliabiliteetti* tarkoittaa luotettavuutta, esimerkiksi tutkimuksen kykyä tuottaa toistettavia tuloksia samalla koeasetelmalla. *Validiteetti* eli pätevyys on kyky mitata juuri sitä, mitä tutkimuksissa tarkalleen halutaan tutkia (Hirsjärvi et al. 2007; Hirsjärvi & Hurme 2008).

Tapaustudkimuksen ja haastattelujen näkökulmasta reliabiliteettia parantaa se, että tutkimusprosessi on dokumentoitu. Laadullisessa aineiston analyysissä on keskeistä luokitte-  
lujen tekeminen, joka tässä tutkimuksessa tehtiin pääteemoittain (Hirsjärvi et al. 2007).  
Tässä työssä tutkimuksesta dokumentointiin tutkimusstrategian valinta ja kuvattiin sen  
käytäntöön pano. Teemahaastatteluista kerrottiin suunnitteluperusteet, haastattelujen to-  
teutus, analysointi ja tulokset. Johdetut toimenpiteet palvelun kehittämiseen on tuotu esiin



ja perusteina on viitattu kulloinkin tulkittavaan teoreettiseen viitekehykseen, joka koostui alan kirjallisuudesta ja tutkimustiedosta.

Toistettavuuden osalta teemahaastatteluissa ei voida kuitenkaan olettaa, että jatkossa saataisiin täysin samoja vastauksia tutkittavan palvelun henkilöiden ja liiketoimintakontekstien muuttuessa. Hirsjärvi & Hurme (2008) kuvaakin, että laadullisessa tutkimuksessa reliabiliteetti koskeekin enemmän tutkijan toimintaa kuin itse vastauksia. Kirjoittajat muistuttavat, että haastattelujen tulos on aina haastateltavan ja haastattelijan yhteistoimintaa.

Tässä tutkimuksessa haastattelijaroolissa toimiminen toi matkan varrella opinnäytetyön tekijälle myös lisää kokemusta, joten lähtökohtaisesti kokeneen haastattelijan käyttäminen heti alusta tasalaatuisena olisi voinut parantaa haastattelutulosten sisältöä, esimerkiksi ohjaamalla keskustelua vielä tiiviimmin teema-alueisiin. Reliabiliteetin kannalta tutkijan toimiminen tutkittavassa kohteessa voi tuoda myös haittaa tutkimukselle, esimerkiksi ohjaamalla sisältöä tiedostamattomasti tai asenteellisesti muuhun kuin tarkoituksenomaiseen suuntaan tutkimuksen kannalta. Tämä tiedostaen, ja haittojen välttämiseksi, haastattelijalla pyrki pysymään hyvin objektiivisena haastatteluissa eikä tuonut esiin omia näkemyksiään. Toisaalta tutkijan kokemus palvelussa työskentelystä toi tehokkaan mahdollisuuden ohjata haastattelua teemojen kautta vielä tarkentavampiin ja syvällisempiin kysymyksiin.

Validiteetin käsite reliabiliteetin tapaan on alun perin peräisin kvantitatiivisesta tutkimuksesta (Hirsjärvi & Hurme 2008). Haastattelujen kohderyhmän valinnan osalta voidaan todeta validiteetin olevan erittäin hyvä, koska tutkimus kohdistui juuri tiettyyn palveluun, haastateltiin juuri tätä kohderyhmää ensisijaisesti suoraan. Haastatteluun saatiin jokaisesta tiimistä kokenut haastateltava. Haastattelujen määrällä saatiin esiin myös saturatiota, eli tietyissä teemojen asioissa vastaukset alkoivat olla toistensa kaltaisia. Tutkimusstrategian pätevyys varmistettiin tukeutumalla alan kirjallisuuteen, jonka tarkemmat perustelut esitettiin tutkimusstrategian kuvauksessa.

Yksi mainittu validiteettia lisäävä tekijä on myös varmistaa lähteiden luotettavuus. Haastatteluihin valitut valittiin mahdollisimman ammattitaitoisista henkilöistä, jotta saatavat näkemykset olisivat laajoja ja syväluotaavia. Haastattelijalla eli opinnäytetyön tekijällä oli kokemusta palvelusta ja sen kontekstista, jolla pystyi myös osaltaan varmistamaan sen, ettei vastauksissa pyritä kaunistelemaan tekemisen prosesseja, vaan kerrotaan miten asiat oikeasti nähdään ja mitä epäkohtia on havaittu. Tutkimusteemojen validiteettia tuki kirjallisuus, löydetty vastaavan kaltaiset tutkimukset ja myös tutkimuksen tekijän pitkä kokemus käytetyistä menetelmistä toimialalla.

Validiteettia voidaan parantaa menetelmätriangulaatiolla. Tällöin vain yhdellä menetelmällä, eli haastatteluilla saaduilla asioilla ja niiden johtopäätöksillä, haetaan vahvistusta muista lähteistä. Tämän onnistuessa voidaan todeta näkemyksen saaneen vahvistusta ja yksimielisyyden. Haastattelutulosten tulkinnassa ja palvelun kehittämisessä vahvistusta

haettiin myös muiden haastateltavien vastauksista, mutta erityisesti teoreettisesta viitekehystä.

Tutkimusasetelman validiteetin jakoja on tutkimuskirjallisuudessa useita. Yhtenä jakona voidaan pitää *sisäiseen* ja *ulkoiseen validiteettiin* jakoa. Sisäisessä validiteetissa tarkastellaan, onko muuttujien suhteen oikeasti kausaalisuus, vai voiko olla vielä useampi huomaamaton tekijä. Laadullisen tapaustutkimuksen osalta voidaan lähinnä tarkastella vain tutkimusstrategian tarkoituksenmukaisuutta, joka tässä työssä jo käsiteltiin myös reliabiliteetin kohdalla (Hirsjärvi & Hurme 2008). Ulkoinen validiteetti on tutkimuksen yleistettävyyttä laajempiin tapauksiin. Ulkoisen validiteetin täyttäminen on tässä tutkimuksessa haasteellista, koska tutkitaan nimenomaan yksilöllistä palvelua ja sen toimintaympäristöä. Toisaalta, löydetty parannukset palvelua kehittäessä ovat kirjallisuuden perusteella toimineet muuallakin, mutta tätä ei vielä voida pitää tämän tutkimuksen yleistettävyyden mittarina. Näitä vertailuja muihin tutkimuksiin, on käsitelty työn tuloksia esittelevässä alaluvussa keskusteluna.

Tutkimuksen haastattelujen seurauksena ja tuloksena tehtyjen yksittäisen kehitystoimenpiteen vaikuttavuutta on myös vaikea tai mahdoton arvioida. Esimerkiksi tiimeissä otettiin retrospektiivit käyttöön melko uutena työkaluna. Tätä ennen ei ollut jatkuvaa toiminnan parantamisen metodia käytössä. Tällöin jo itsessään niillä voidaan parantaa olemassa olevien prosessien laatua, vaikka itse prosessia ei muutettaisi. Tämän takia usean toimintaa kehittävän rinnakkaisen toimenpiteen myötä on vaikea sanoa mikä niistä oli osaltaan ratkaisevin ja vaikuttavin.

Kriittiseen tarkasteluun on syytä nostaa myös se, voidaanko lyhyellä tutkimusajalla vielä todentaa riittävän hyvin kehitystoimenpiteiden pitkän aikajänteen parannukset. Välitöntä hyötyä saatiin työtehtävöiden näkyvyydestä ja hallinnasta, mutta todelliset vaikutukset mittaroituvat varsinkin tiimityön kehittämisessä vasta pidemmän seurantajakson jälkeen. Tiimien muuttuminen aidosti itseohjautuvaksi on kirjallisuuslähteiden perusteella vuosien muutosprosessi, joten tämän pohjalta lyhyessä tutkimusajassa ei voi tehdä pitkälle meneviä johtopäätöksiä.

Tämän tutkimuksen tavoitteiden voidaan kuitenkin sanoa täyttyneen sille varatun ajan, rajausten ja resurssien puitteissa. Tutkimuksen ollessa dokumentoitu ja tuettu teoreettisella viitekehyksellä, voidaan sanoa tämän tutkimuksen olevan luotettava ja pätevä.

## 5.4 Jatkotutkimuksen aiheita

Kuten edellä esitetyissä alaluvuissa todetaan, valitut tehtävienhallinnan menetelmiin pohjautuneet tekijät ja ominaisuudet löydettiin, jonka myötä muutoksia alettiin ottaa kohdepalvelussa myös käyttöön. Aihepiiri on kuitenkin laaja ja jatkotutkimuksen potentiaalia on moneen eri liitännäiseen aiheeseen. Jokaisesta annetusta kehitystoimenpiteestäkin

voisi jo itsessään tehdä oman tutkimuksen. Seuraavaksi kuvataan selkeät jatkotutkimuksen aiheet, jotka tukevat tämän tutkimuksen aihepiiriä, tai nousivat esiin liittyen läheisesti tehtävienhallintaan, mutta rajautuivat tämän tutkimuksen sisällöstä ulos.

#### *Tiimien rakenteet ja toiminta*

Tämä tutkimus painottui tehtävienhallinnan kehittämiseen eikä syvällisesti tiimitoiminnan perusteisiin ja sen kehittämiseen. Tehtävienhallinnan keskiössä on kuitenkin väistämättä tiimi ja sen tehokas toiminta, kuten tutkimuskysymykseen vastatessa todettiin. Kohdepalvelun käytännön kehityssuosituksissa tiimien kokoonpanoihin ja rakenteisiin annettiin tehtävienhallintaa tukevat suositukset. Tehokkaan ja toimivan tiimijaon luominen palveluun vaatii siihen vaikuttavien tekijöiden tunnistamisen, keskinäisen analysoinnin ja käytäntöön panon, joka itsessään on jo oma laaja kokonaisuutensa ja muodostaisi mahdollisen tutkimuskokonaisuuden.

#### *Työtehtävävirtojen kehitys Lean-menetelmin*

Työssä ei otettu kantaan yksittäisten tehtävävirtojen sisältöön vaan keskityttiin niiden visualisointiin, sovittamiseen ja hallitsemiseen esimerkiksi Kanban-menetelmällä. Haastattelussa tuli ilmi, että tehtävävirroissa koettiin olevan hukkaa ja tarpeettomia aktiviteetteja, joiden poistoa kannattaisi tutkia. Tähän Lean-kirjallisuus tarjoaa työkaluja läpikävelyn ja arvovirtakaavioiden mallinnuksen kautta. Näiden virtojen tutkiminen, analysointi ja hukan poisto yksityiskohtaisesti voisi olla oma tutkittava kokonaisuutensa.

#### *Kanbania käyttävien tiimien väliset tehtäväriippuvuudet*

Tässä tutkimuksessa päädyttiin tehtävienhallintaa kehittämään Kanban-menetelmän avulla. Kohdepalvelussa ei ollut merkittäviä riippuvuuksia tiimien kesken, joten riippuvuuksien hallitseminen ei noussut tässä tutkimuksessa erityisen ratkaistavaksi. Haastattelussa nähtiin kuitenkin kehityssuunta, jossa osa kehitystehtävistä kosketti useampaa tiimiä ja organisointia tehtiin perinteisen projektinhallinnan riippuvuuksien kautta.

Työn kirjallisuuskatsauksen yhteydessä esiteltiin lyhyesti tuotekehityksen skaalaamiseen soveltuvat menetelmät, kuten esimerkiksi esitelty SAFe-viitekehys, jossa laajan tuotekehityksen eri osakokonaisuuksia kehittävät eri tiimit. Jos tiimien keskinäisten riippuvuuksien hallinta esimerkiksi laajoissa kehitystöissä laajenee palvelussa merkitsevämmäksi, tulee oleelliseksi tutkia, miten Kanban-mallissa sovitetaan eri tiimien toiminta synkronoidusti ja ennustettavasti yhteen.

#### *Tässä tutkimuksessa esitetyt kysymykset laajemmassa kontekstissa*

Tehtävienhallinnan ongelmakenttä on todennäköisesti samankaltainen muissakin tietotekniikan palveluyrityksen tiimeissä asiakkuuden erityispiirteistä huolimatta. Tämän työn jatkoksi voisi tutkia tehtävienhallintaa laajemmassa kontekstissa, eli miten yleistettäviä

löydetyt parannukset ja ominaisuuksien tarkemmat valintaperusteet olisivat, kun kohteena ei olisi enää yksittäinen yritys tai asiakasympäristö. Tämän ja vertailtavien muiden tutkimusten perusteella, Kanban yhdistettynä ketterillä menetelmillä on lähinnä oikeaa mallia tiimien tehtävien hallintaan, jossa erityisesti aikataulutetun kehityksen ohella tehdään ennustamatonta ja nopeampaa häiriönhallintaa. Tästä ei löytynyt kuitenkaan laajoja yleistäviä tutkimuksia.

## LÄHTEET

- Al-Baik, O. & Miller, J., 2015. The kanban approach, between agility and leanness: a systematic review. *Empirical Software Engineering*, 20(6), pp.1861–1897. Available at: <https://doi-org.libproxy.tut.fi/10.1007/s10664-014-9340-x>.
- Andersson, D.J., 2010. *Kanban - Successful Evolutionary Change for Your Technology Business*, Washington.
- Barrick, M.R. et al., 1998. Relating member ability and personality to work-team processes and team effectiveness. *Journal of Applied Psychology*, 83(3), pp.377–391. Available at: <http://psycnet.apa.org/record/1998-02893-003>.
- Beck, K. & Andres, C., 2006. *Extreme Programming Explained 2nd edit*, Addison-Wesley.
- Bell, S.T., 2007. Deep-level composition variables as predictors of team performance: A meta-analysis. *Journal of Applied Psychology*, 92(3), pp.595–615. Available at: <http://psycnet.apa.org/record/2007-06438-001>.
- Blom, R., Melin, H. & Pyöriä, P., 2001. *Tietotyö ja työelämän muutos: palkkatyön arki tietoyhteiskunnassa*, Gaudeamus.
- Bond, C.F. & Titus, L.J., 1983. Social facilitation: A meta-analysis of 241 studies. *Psychological bulletin*, 94(2), pp.265–292.
- Cohen, S.G., Ledford, G.E. & Jr, C.S.G., 1994. The Effectiveness of Self-Managing Teams: A Quasi-Experiment. *Human Relations*, 47(1), pp.13–43. Available at: <http://journals.sagepub.com/doi/10.1177/001872679404700102>.
- Eskola, J. & Suoranta, J., 1998. *Johdatus laadulliseen tutkimukseen*, Tampere: Vastapaino.
- Goldratt, E., 1984. *The Goal*,
- Goncalo, J.A., Flynn, F.J. & Kim, S.H., 2010. Are Two Narcissists Better Than One? The Link Between Narcissism, Perceived Creativity, and Creative Performance. *Personality and Social Psychology Bulletin*, 36(11). Available at: <http://journals.sagepub.com/doi/10.1177/0146167210385109>.
- Hackman, J.R., 1987. The design of work teams. *Handbook of organizational behavior*, pp.315–342.

- Hackman, J.R., 1986. The psychology of self-management in organizations. *The Master lectures, Vol. 5. Psychology and work: Productivity, change, and employment*, pp.89–136. Available at: <http://psycnet.apa.org/record/1986-97965-003>.
- Haikala, I. & Mikkonen, T., 2011. *Ohjelmistotuotannon käytännöt*, Helsinki: Talentum.
- Highsmith, J., 2001. History: The Agile Manifesto. Available at: <http://agilemanifesto.org>.
- Hines, P. & Rich, N., 1997. *The seven value stream mapping tools*,
- Hirsjärvi, S. & Hurme, H., 2008. *Tutkimushaastattelu*, Gaudeamus.
- Hirsjärvi, S., Remes, P. & Sajavaara, P., 2007. *Tutki ja kirjoita*, Tammi.
- Ikonen, M., 2011. *Lean Thinking in Software Development: Impacts of Kanban on Projects*. University of Helsinki. Available at: <http://hdl.handle.net/10138/28453>.
- ITIL, 2011. *IT Infrastructure Library*, The Stationery Office (TSO).
- Kantolahti, T. & Tikander, T., 2010. *Puheenvuoroja työn kuormittavuudesta*, Helsinki. Available at: <http://urn.fi/URN:ISBN:978-952-00-3011-7>.
- Kilad, C. & Kanfer, R., 2006. Toward a Systems Theory of Motivated Behavior in Work Teams. , 27(Research in Organizational Behavior), pp.236–237. Available at: <https://www.sciencedirect.com/science/article/pii/S0191308506270060>.
- Kleingeld, A., van Mierlo, H. & Arends, L., 2011. The effect of goal setting on group performance: a meta-analysis. *Journal of Applied Psychology*, 96(6), pp.1289–1301. Available at: <https://www.ncbi.nlm.nih.gov/pubmed/21744940>.
- Kniberg, H., 2011. *Lean from the Trenches: Managing Large-Scale Projects with Kanban*, The Pragmatic Programmers.
- Kniberg, H. & Skarin, M., 2010. *Kanban and Scrum - Making the Most of Both*, C4Media, publisher of InfoQ.Com. Available at: <https://www.infoq.com/minibooks/kanban-scrum-minibook>.
- Kozlowski, S.W.J. & Bradford, S.B., 2013. Work Groups and Teams in Organizations. In N. W. Schmitt, S. Highhouse, & I. B. Weiner, eds. *Handbook of Psychology (Vol. 12): Industrial and Organizational Psychology*. Hoboken, New Jersey.: John Wiley & Sons, Inc., pp. 412–468.
- Ladas, C., 2008. *Scrumban and other essays on Kanban Systems for Lean Software Development*, Seattle: Modus Cooperandi.

- Langfred, C.W., 2004. Too Much of a Good Thing? Negative Effects of High Trust and Individual Autonomy in Self-Managing Teams. *Academy of Management Journal*, 47(3), pp.385–399. Available at: <http://psycnet.apa.org/record/2004-15942-006>.
- Larman, C., 2004. *Agile and Iterative Development: A Manager's Guide*,
- Lawler, E.E., 1986. *High involvement management*, San Francisco. Available at: [https://books.google.fi/books/about/High\\_involvement\\_management.html?id=OB5PAAAAMAAJ&redir\\_esc=y](https://books.google.fi/books/about/High_involvement_management.html?id=OB5PAAAAMAAJ&redir_esc=y).
- Leffingwell, D., 2019. About Scaled Agile Framework. *Scaled Agile Inc.* Available at: <https://www.scaledagileframework.com/about/> [Accessed January 27, 2019].
- Leffingwell, D., 2008. *Scaling Software Agility: Best Practices for Large Enterprises (Agile Software Development Series)*, Pearson Education.
- Lehti, M., Rouvinen, P. & Ylä-Anttila, P., 2012. *Suuri hämmennys: Työ ja tuotanto digitaalisessa murroksessa*, Helsinki: Taloustaito Oy.
- Maassen, O. & Sonneveld, J., 2010. Kanban at an insurance company (Are you sure?). *11th International Conference, XP 2010: Agile Processes in Software Engineering and Extreme Programming*, pp.297–301.
- Manz, C.C. & Sims, H.P.J., 1987. Leading Workers to Lead Themselves: The External Leadership of Self- Managing Work Teams. *Administrative Science Quarterly*, 32(1), pp.106–129. Available at: [https://www.jstor.org/stable/2392745?seq=1#page\\_scan\\_tab\\_contents](https://www.jstor.org/stable/2392745?seq=1#page_scan_tab_contents).
- Metsä-Tokila, T., 2017. *Kasvun mahdollistajat - toimialaraportti ohjelmistoalasta ja teknisestä konsultoinnista*, Helsinki.
- Michal, P. & Paliszkievicz, J., 2015. Framework of Strategic Learning: The PDCA cycle. *Management 10* (2), pp.149–151.
- Modig, N. & Åhlström, P., 2013. *Tätä on Lean*,
- Neuman, G. & Wright, J., 1999. Team effectiveness: beyond skills and cognitive ability. *Journal of Applied Psychology*, 84(3). Available at: <https://www.ncbi.nlm.nih.gov/pubmed/10380418>.
- Nikitina, N., Kajko-Mattsson, M. & Strale, M., 2012. From Scrum to Scrumban: a case study of a process transition. *2012 International Conference on Software and System Process, ICSSP 2012 - Proceedings*.
- O'Leary-Kelly, A.M., Martocchio, J.J. & Frink D., D., 1994. A review of the influence of group goals on group performance. *Academy of Management Journal*, 37(5),

- pp.1285–1301. Available at: <http://psycnet.apa.org/record/1995-09378-001>.
- Press, G., 2017. 5 Top Technologies for Digital Disruption. *Forbes*. Available at: <https://www.forbes.com/sites/gilpress/2017/04/27/5-top-technologies-for-digital-disruption/#31a876c14898> [Accessed May 27, 2018].
- Prewett, M.S. et al., 2009. The team personality–team performance relationship revisited: The impact of criterion choice, pattern of workflow, and method of aggregation. *Human Performance*, 22(4), pp.273–296. Available at: <http://psycnet.apa.org/record/2010-11742-001>.
- Royce W., W., 1970. *Managing the development of large software systems*,
- Schwaber, K. & Sutherland, J., 2017. *The Scrum Guide™, The Definitive Guide to Scrum: The Rules of the Game*, Available at: <https://www.scrumalliance.org/learn-about-scrum/the-scrum-guide> [Accessed September 19, 2018].
- Sendil Kumar, C. & Panneerselvam, R., 2007. *Literature review of JIT-KANBAN system*,
- Shepperd, J.A., 1993. Productivity Loss in Performance Groups: A Motivation Analysis. *Psychological bulletin*, 113(1), pp.67–81. Available at: <https://people.clas.ufl.edu/shepperd/files/Loafing-PsychBull.pdf>.
- Steinberg, R., 2011. *Best Management Practice: ITIL Service Operation* 2nd ed., United Kingdom: The Stationery Office (TSO).
- Stewart, G.L. & Barrick, M.R., 2004. Four Lessons Learned From the Person-Situation Debate: A Review and Research Agenda. *EA's organization and management series. Personality and organizations*, pp.61–85. Available at: <http://psycnet.apa.org/record/2004-12250-003>.
- Stewart, G.L. & Manz, C.C., 1995. Leadership for Self-Managing Work Teams: A Typology and Integrative Model. , 48(1), pp.747–770. Available at: <http://journals.sagepub.com/doi/10.1177/001872679504800702>.
- Torkkola, S., 2015. *Lean - asiantuntijatyön johtamisessa*,
- Tuckman, B.W., 1965. Developmental sequence in small groups. *Psychological bulletin*, 63(6), p.384. Available at: <http://psycnet.apa.org/record/1965-12187-001>.
- Tupes, E.C. & Christal, R.E., 1961. Recurrent Personality Factors Based on Trait Ratings. *Technical Report ASD-TR-61-97*.
- VersionOne, 2018. The 12th annual State of Agile report.
- Wang, X., Conboy, K. & Cawley, O., 2012. “Leagile” software development: An



experience report analysis of the application of lean... *The Journal of Systems and Software*, 85.

Watson, W.E., Kumar, K. & Michaelsen, L.K., 1993. Cultural diversity's impact on interaction process and performance: Comparing homogenous and diverse task groups. *Academy of Management Journal*, 36, pp.590–602.

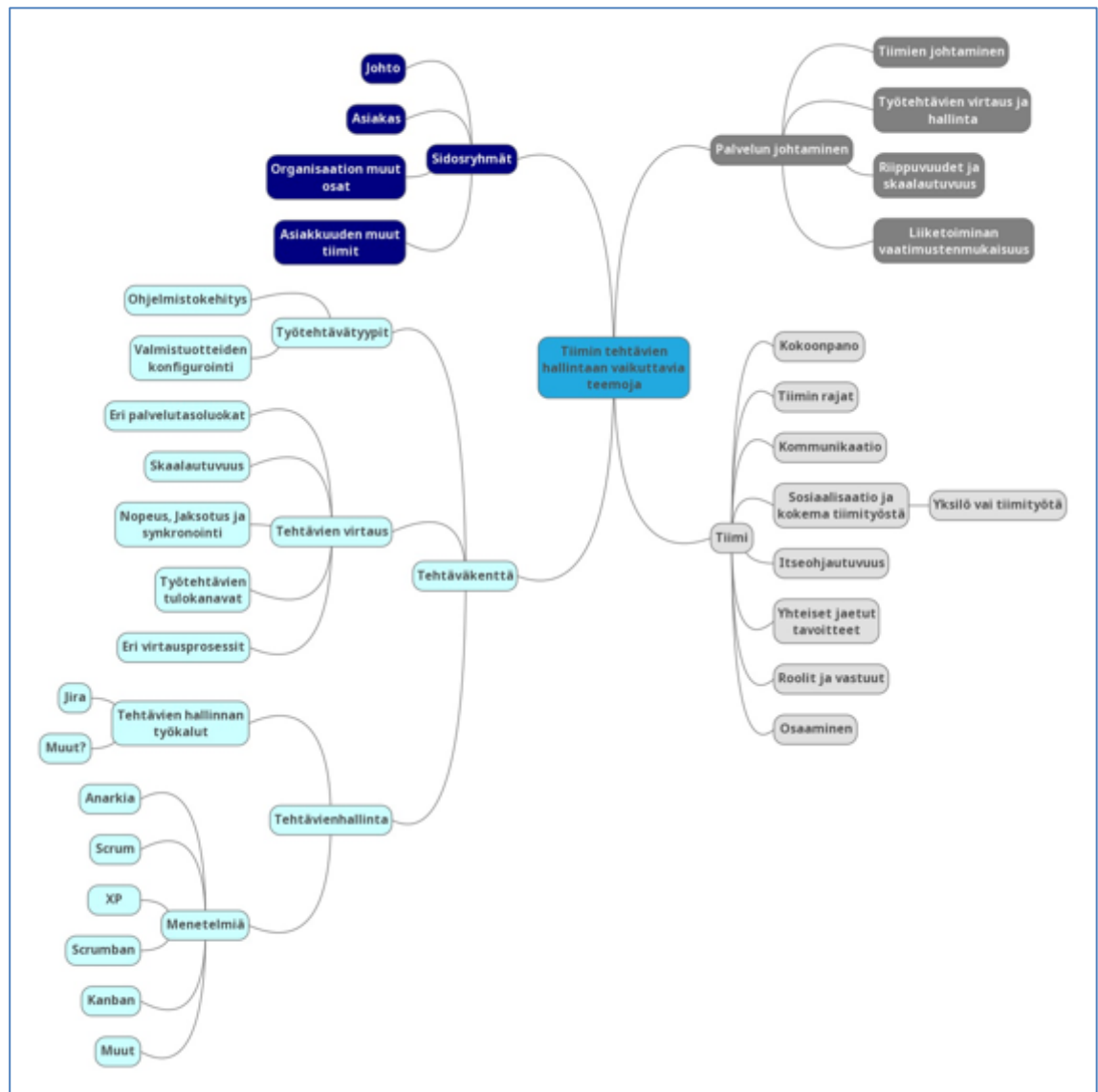
West, M.A. & Anderson, N.R., 1996. Innovation in Top Management Teams. *Journal of Applied Psychology*, 81(6), pp.680–693. Available at: [https://www.researchgate.net/profile/Neil\\_Anderson7/publication/278897738\\_Innovation\\_in\\_top\\_management\\_teams/links/56e96ccd08ae77f87278fdf3/Innovation-in-top-management-teams.pdf](https://www.researchgate.net/profile/Neil_Anderson7/publication/278897738_Innovation_in_top_management_teams/links/56e96ccd08ae77f87278fdf3/Innovation-in-top-management-teams.pdf).

Womack, J.P., Jones, D.T. & Roos, D., 1991. *The Machine That Changed the World: The Story of Lean Production*,

Yin, R.K., 1984. Case study research: Design and methods, (Beverly Hills, CA).

Zajonc, R.B., 1965. Social facilitation. *Science*, 149, pp.269–274. Available at: [http://www.psychwiki.com/wiki/Zajonc,\\_R.\\_B.\\_\(1965\).\\_Social\\_facilitation.\\_Science,\\_149,\\_269-274](http://www.psychwiki.com/wiki/Zajonc,_R._B._(1965)._Social_facilitation._Science,_149,_269-274).

## LIITE 1: MIELLEKARTTA HAASTATTELUN PÄÄTEEMOISTA



**Kuva 9** Miellekartta, jota käytettiin apuna teema-alueiden löytämiseen sekä haastattelun ohjaamiseen. Neljä pääteemaa jaettuna myös vielä syvempiin aliteemoituksiin, joiden perusteella voitiin muodostaa myös tarkkoja kysymyksiä.